Using Deep Linguistic features to predict Depression

Thang Nguyen
University of Maryland
daithang@umiacs.umd.edu

William Armstrong
University of Maryland
armstrow@umd.edu

Anilesh Shrivastava
University of Maryland
anilesh@umd.edu

Abstract

We participated in an in class shared task of predicting CES-D Depression Score by using linguistic features on Facebook Statuses of the users. Our attempt was to look beyond surface features like bag of words and Topic model. To that end we explored several off the shelf tools for parts of speech tagging, Supervised LDA, deep learning and active learning. In this paper we document our experience of using such tools, and challenges we faced while building this prediction system. In the end we were able to achieve good results through supervised LDA and got some promising insight into deep learning.

1 Introduction

With millions of people suffering from depression, there is substantial room for improvement in our ability to not only diagnose but also understand the patterns of this ailment. The clinical prognosis is often made by observing potential candidates of depression in various situations. This often involves incorporating voluntary responses from the candidate to a range of questions, either written or answered in an interview.

One such screening method is the Center for Epidemiological Studies Depression Scale (CES-D). This scale gives a cursory view of person's depression level. This self-reported scale is based on responses to 20 questions where each question is a measure of some symptoms of depression. A respondent has to answer these questions on a numerical scale of 1-4, which is a measure of frequency for an event of interest. For example a question may be about mood changes, sleeplessness, tiredness etc. What results is an encoding of candidate's mental condition as a vector representation in the question space. A final CES-D score is calculated by summing up the individual values.

Our shared task in the project was to study Facebook status messages of a set of users and engineer linguistics-based features that represent the effects of a person's state of mind on his language. This is becoming a highly desirable and convenient method for detecting psychological patterns of the users, especially considering the current abundance of social media use. This large amount of user-generated text encodes a person's psychological finger print and our task is to decode this to reveal the cognitive interpretation of a person's experiences.

We correlated the linguistic features we extracted from the Facebook texts with CES-D scores and question responses, essentially mimicking the cognitive process of a person while answering these questions. We assume that while answering these questions the user maintained utmost objectivity and even though veracity of the responses cannot be verified, our model incorporates enough variation that such outliers will not be significant. Hence we are trying to assert through linguistic inquiry and information extraction what the response to a particular question asked in the CES-D Questionnaire would be. In the end we classify each user as depressed or otherwise based on these predicted responses.

We draw upon our experiences from the Computation Linguistic -1 Course in fall 2013 in which we attempted to predict a big 5 personality trait using Facebook statuses, as an extension to a similar work done by (Resnik et al., 2013) on Stream of consciousness essays. During this task we mainly extracted surface features from the data, producing models to incorperate such features as bag of words, sentiment-lexicon based features, and Topic modeling. For this project, we reused and re-tuned most of the text processing tools developed during the previous project.

Our best results were achieved through use of a supervised variation of Latent Dirichlet Allocation, known as Labeled LDA (LLDA), as implemented by Stanford University in the Stanford Topic Modeling Toolbox (Ramage et al., 2009), as well as its extension, known as Partially Labeled Dirichlet Allocation (PLDA), by the same group(Ramage et al., 2011). We explored the possibility of using the label inference features of the model to flat out predict whether a user was depressed or not, but obtained better results by simply predicting the question responses and using a separate model to turn those into a prediction of depression.

Inspired by Mohit's Work, we tried to learn the deep representation of words and then use this representation for prediction task. This did not result in the kind of success we were hoping for but our own understanding of such models was enhanced and it provides potential for future improvements into this kind of task. This project also allowed us to experiment with the paradigm of Active learning using human annotators. This task was limited by the ability of the group members to annotate, but it has potential to improve the task, given adequate resources.

In section 2 we will describe the data set, our initial analysis of the data, and preprocessing. Sections 3 through 6 will describe in detail our specific models and their corresponding features that we developed as well as the ML algorithms we used for prediction and regression. Section 3 focuses on the simple baseline starting point, section 4 discusses our attempt to incorperate deep learning, and section 4 mentions our experimentation with active learning and its effects. Finally, section 6 goes into much greater detail and analysis about our most successful techniques involving LLDA, which will be followed with overall results and anlysis in sections 7 and 8.

2 Data and Approach

Our data set we used for training our models consisted of 700 users and around 1 and a quarter million status messages, with date and time of posting. We also have Meta data related to each users, identifying their gender, their question-wise score on the CES-D questionnaire, the date and time on which the test was taken, the total CES-D score, and the classification as depressed or not. The cutoff on total score to classify a person as depressed is 33.

The status message data needed lot of cleaning before we could apply any linguistic tools. As expected for informal social media, it contained many errors including misspelled words, abbreviations, unnecessary punctuations, emoticons, numbers, foreign languages etc.

We began by removing punctuation, emoticons, and stop words. We had used both punctuation and emoticons as features last semester but we did not observe a significant signals from these features, hence we ignored them for this task. We concatenated all the status messages belonging to one user and created single record for that user in our final input dataset.

We ran an analysis of questions scores to see the contribution of scores in the classification, using a chi-squared test. We observed that some of the questions did not contribute significantly, like question 12, and removing them did not change the final classification much. We also analyzed the classification accuracy we can get by just considering top k questions ranked by our analysis. We did this to understand the importance of individual questions in our task, and if it is possible to predict the individual question scores on the test data. We determined that information from the questions could indeed be used to predict a user's classification by using methods that accounted for more than just their summed values.

3 Baseline Model using surface features

We constructed our baseline from where we ended in the last semester.

3.1 Bag of Words

Our bag of words features contained a TFIDF vector for the top 1000 words found in the input. We built a vocabulary from training set and composed the final files for train and test from these 1000 words.

3.2 LIWC

We also used Pennebaker and King‘s Linguistic inquiry Word count Lexicon Dictionary(James W. Pennebaker and Francis., 2007). The LIWC Dictionary contains 64 categories, such as ‘sad ’, ‘anger’,‘positive emotion’,‘negative emotion’etc.

We represented each status message as a distribution over these categories and for each user we summed the values, normalizing by length of the messages and the number of status messages, hence forming 64 features–one each for a category.

3.3 Topic Model using LDA

Using Mallet, we obtained a topic model for our corpus, where each user is taken as a single document containing all their concatenated status messages. We obtained 40 topics from LDA and represented each user as a distribution vector over these 40 topics. These 40 elements were used as additional features in our baseline model. We obtained the baseline by combining 1000 bag of words + 64 LIWC + 40 Topic model features. Experimentally, we first ran this as a classification problem and then as a regression problem over the total CES-D scores. We observed that regression scores were slightly better than the classification score on the baseline.

3.4 Flat Parts of speech Tagging

For a deeper linguistic analysis of the status messages, we decided to extract features that capture grammatical nuances of the users and how they may be affected by the state of mind of the user. This task proved particularly difficult as many statuses are malformed and do not use conventional vocabulary or grammar. Furthermore, Facebook status messages are often incomplete sentences and have varying degrees of ambiguity.

We found a good off the shelf tool in tweet NLP (evin Gimpel and Smith, 2011), which specializes in tagging small text messages seen online. This tool worked on the majority of the statuses and returned a tag for each word. It seemed to understand the idioms of the *social network.* The tags produced by the tools were used as bag of tags (TFIDF vectors). This technique did not produce very good result on cross validation, but it managed to improve the baseline by a small margin. We suggest improvement on this by using bigram features from our sequence of tags.

4 Deep Learning

Our initial plan was to apply deep learning to learn a sentiment compositionality (Mohit et al., 2014), from each Facebook status. To do that we need a dependency parse tree for each status message. We can try running a probabilistic parser (like Stanford parser), but as seen in Section 3.4, it may not yield the correct tree for a status. Due to general difficulties discussed in 3.4, here also we decided to use flat tree. Hence the sentiment of a status is the combination of sentiment of individual words. We also would like to learn the deep representation of words and then use these representation for prediction task .

4.1 Tool & data

We were able to reuse code that Mohit used for deep learning to investigate the political ideology problem. We modified the code to make it work with a flat tree. As per Mohit's suggestion, we set the dimension of word vector to be 100, the batch size is equal to the number of training instances divided by 25. We would run 1000 iterations per batch job. The system took around 10 hours to finish training, and around 5 minutes to output the vector representation for each status from the Test set.

The algorithm outputs a 100-dimension vector for each word in a status. We average all word vectors for a particular status to get a 100-dimension vector for it. Finally, for each user we take an average of vectors corresponding to all his statuses, hence getting a 100-dimension vector encoding of the user.

4.2 Results

Running an SVM algorithm using vectors of representation for users as features, we are able to get and F score of 0.37 using 10-fold cross-validation. However, applying this method on the Test data from Kaggle gave us a very low F score of 0.24.

5 Active learning approach

After analyzing Facebook statuses in our dataset, we realized that they contain so much noise that it may severely affect any supervised algorithms. For example, statuses like 'I hate being alone' or ' I am tired' are related to depression but statuses like 'France is a big country' have nothing to do with depression. We think that understanding what causes depression and what signals are associated with depression are critical before using any algorithm. Furthermore, we can use our own intuition to scan a Facebook status and pinpoint whether the status contains depression related signals or not. So we designed a simple annotation task that serves the purpose of distinguishing negative emotional signals from other signals.

We decided to use Dualist, an active learning tool, for annotation. *Active learning* helped reduce the number of statuses needing to be labeled by a large number.

5.1 Annotation

The task uses two labels, *'positive'* for depression related statuses and *'negative'* for the rest. We randomly selected only five statuses per user. This composed 3500 Facebook statuses for the annotation task. Dividing the task into several subtasks, we used Dualist to annotate 30 statuses, after which we used Dualist to do retraining and then continue to another subtask.

Two human annotators participated in the task. After each retrain of the model (subtask) the annotator checks the classification results from Dualist and manually checks how the model assigns probability on some unlabeled instances to decide when to stop labeling. This process was rather manual and heuristic but we stopped labeling when each participant annotated around 1,200 statuses. Taking the intersection of these two sets, we were able to get 178 *'positive'* statuses, and 828 *'negative'* statuses, making a total of 1,006 statuses.

To evaluate the quality of annotation, we used deep learning as in section 4 to train the model on these 1,006 statuses. We then used the classification results for each status to run regression and achieved a very high F-Score =0.97 on 10-fold cross validation.This confirmed our intuition about the distinction between depression related statuses with those that are not. The deep learning produced the classification for a status in the form of probability vector. For example, the status 'I hate vacation' is mapped to [0.31, 0.69] where the first entry reflects the 'negative signal' in this status. The regression on these probability vectors gives a cut-off value of 0.252. In other words, if a status is depression related, the first entry in its probability vector is likely to be larger than 0.252.

5.2 Filtering noises

After annotation and evaluation of the quality of annotated statuses, we again used the model learned from deep learning to learn the probability vector for every Facebook status from the dataset. Using 0.252 as the cut-off threshold gives 10,970 depression related statuses. Because there are some users that contain no status with the first entry larger than 0.252, the number of users for negative statuses is 620 instead of 700.

5.3 Results

We incorporated our LLDA model onto the 620 users that have at least one status related to depression as we defined above. The 10-fold cross validation F-Score for this model is 0.803. Doing prediction on test data from Kaggle, we got a lower-than-expected 0.39 for F score.

6 Labeled Latent Dirichlet Allocation

6.1 Tool & Data

After failing to find a suitable implementation of Supervised LDA that could provide a regression value for user scores, we chose to instead use Labeled LDA with each user labeled as either positive or negative for depression. We used the Stanford Topic Modeling Toolbox (Ramage et al., 2009) and treated all concatenated statuses as a single "document" for the purpose of the training.

On the preprocessed dataset we additionally removed any terms that appeared in fewer than 4 documents as well as any document that contained less than 5 terms (post filtering). We also removed labels that were found in fewer than 10 documents and the 30 most common terms from the final vocabulary.

6.2 Single-Label LLDA

In our initial experiment a single label (depressed or not) was submitted to the LLDA tool and it returned a probability for each user being labeled with each label. To classify we simply applied a threshold over the resulting probabilities. This was roughly equivalent to Supervised LDA's output of a continuous score for each user. We experimentally tuned the threshold to obtain a desirable precision and recall, the best of which was somewhere around 0.995 due to our chosen emphasis on classifying just the positive instances that made it best to always assume depression unless we were absolutely sure it wasn't.

The predictions at a reasonable threshold for this technique were not very effective (0.36/0.42 for

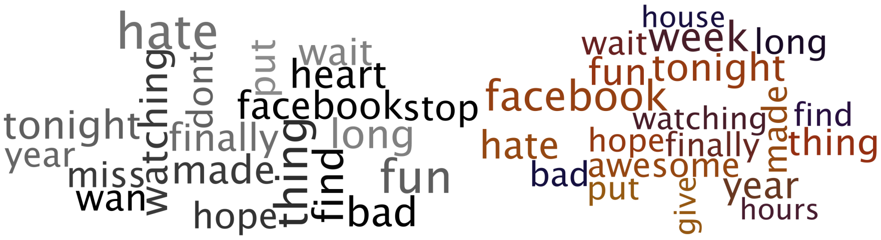

Figure 1: Depressed vs. Non Depressed Top Terms for Label

dev/test at 0.7 threshold). However we can "game" the metric, and get up to the 0.28/0.47 metric by returning as many depressed users as possible with a threshold of 0.995. The top words in the final distribution for classes are shown in Figure 1, perceptually in agreement with our definition of depression ('awesome' and 'fun' vs. 'bad' and 'miss'). However, there is considerable overlap with words like 'hate' and 'hope' unexpectedly appearing in both categories.

These overlaps indicate that perhaps further filtration of words such 'facebook' should be added to the model, since they are common and do not contribute much to classification. One possible explanation for this behavior is that the tool considers both labels as a valid output possibility option, so it is allowed to share terms between states, as would any traditional LDA application (Ramage et al., 2009).

6.3 Multi-Labeled LLDA

Our second and more successful technique was to take advantage of the tool's ability to handle multiple labels per document, as if they were "document tags" in a more traditional LLDA application. We arbitrarily chose to label each user's status with any question that they answered with either of the top half of the options (2 or a 3), which amounted to at least 1 label for most of the training set. Thus we trained the LLDA model to label unseen data with the questions a user would most likely score high on.

We started by verifying the intuition of the resulting topic distributions by reviewing the terms that most contributed to each question and ensuring they correlated with the question descriptions from CES-D. The LLDA implementation also uses the topic models generated to produce a vector of credit attribution, or the percentage that each of its assigned labels was "contributing to" a document (Ramage et al., 2009). These percentages were used as a training feature vector, and the predicted distribution of labels was used for the test-set feature vector.

When we used NaiveBayes predictor over the 20 training attribution scores, we were able to cross-validate the training data with an f-score of 0.388, and predict unseen data (~100 users randomly split from the training set) with an f-score of 0.333.

6.4 Difficulties

The principal drawback of this technique was that feature vectors for each user would all sum to one, since they are interpreted as the probability of each question being a "tag" of the document (Ramage et al., 2009). Additionally, since the model considers all 20 questions as possibilities for the test set and only the provided subset of labels as possibilities for the training set, the probability is more distributed in the test data and the values tend to be smaller. This doesn't quite reflect the nature of the task at hand, since a user who would answer all questions with a high score would have a low-valued feature vector of {1/20, 1/20, ..., 1/20}, for which any given entry will not appear indicative of depression.

Although the machine learning tools appeared to be able to account for this fairly well, better results in some instances were achieved by turning the resulting probabilities back into binary values. Thus any question for which the probability of the label was greater than 1/21 was marked with a 1 and all other columns were marked with a 0 (see Table 2). Using these new feature vectors allowed a small improvement in the NaiveBayes predictive function on the test set. This model also assumes that every user would answer with a high value for at least one question, which was true for all but 2 of the instances in the 700 users of the training set so it didn't seem like an unreasonable assumption to make.

```
User1,1,2,2,1,1,1,0,1,2,1,2,1,1,2,2,2,1,2,2,2,2,30,-
User2,1,3,3,2,0,3,3,1,3,1,3,1,2,1,2,0,3,1,3,1,3,39,+
User3,1,0,3,3,0,0,0,0,0,0,3,0,0,0,0,0,0,3,0,0,12,-
–>
User1,q1 q2 q8 q10 q13 q14 q15 q17 q18 q19 q20, bored tired harassed ...
User2,q1 q2 q3 q5 q6 q8 q10 q12 q14 q16 q18 q20,lonely sicky gal ...
User3,q5 q7 q12 q18,facebook experiment reading don speak post ...
```

Table 1: Conversion of Question Scores to Labels

id	q1	q2	q5	q6	q8	q10	q12	q13	q15	q16	q19	q20	class
U1	0.17	0.46	0.00	0.00	0.00	0.00	0.00	0.10	0.04	0.00	0.06	0.18	-
U2	0.02	0.11	0.04	0.25	0.15	0.25	0.00	0.00	0.00	0.18	0.00	0.00	+
U3	0.00	0.00	0.14	0.00	0.00	0.00	0.86	0.00	0.00	0.00	0.00	0.00	-

U1	1	1	0	0	0	0	0	1	0	0	1	1	-
U2	0	1	0	1	1	1	0	0	0	1	0	0	+
U3	0	0	1	0	0	0	1	0	0	0	0	0	-

Table 2: Binary vs. Continuous Training Feature Vectors (zero columns ommitted)

6.5 Topic Analysis

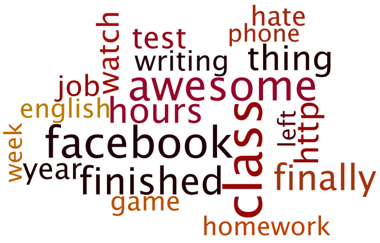

Figure 2: **Question 20:** *I could not get "going"*

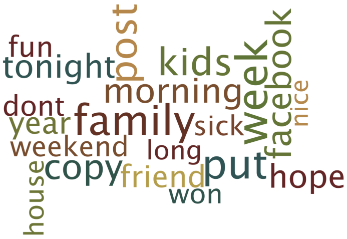

Figure 3: **Question 10:** *I felt fearful*

Figures 2 and 3 show the topics for the two questions which contributed most to the final prediction model. These results are intuitive to the human reader as, for question 20, someone who has trouble getting 'going' may be prone to procrastinate on facebook and mention what they are struggling with in their status ('homework', 'job', 'writing'). They also may mention what they are doing while procrastinating ('game', 'watch', 'phone') or even go so far as to share a link to what they are reading online ('http'). Question 10 is slightly less intuitive, but generally includes words that may be associated with worries ('sick', 'family', 'house') or the progression of time as they project their fears into the future ('tonight', 'week', 'year').

Ironically 'hope' is in this set as well, but it is possible it was frequently negated in the corresponding statuses. As mentioned with the single class labels, we see the word 'facebook' occurring in almost every topic, so it probably should have been filtered out since it doesn't narrow down the topics. However, since the multi-labeling technique does not require that labels be mutually exclusive, as long as there is a few topics that do not include that word, as in our data, it should only serve to narrow down the possible label set.

Finally, we tried to improve the predictive power of our LLDA inference by adjusting some of the parameters to that model (ie removing additional common words, increasing the number of iterations, etc.). Although we saw topic descriptions that were perhaps more coherent due to these adjustments, none resulted in significant improvement to the overall results, so we left further pursuit of these and other hyper parameters for future work.

Model	10-cv Score	Dev Score	Kaggle Score
Class as Label	N/A	0.357	0.471
Questions as Labels	0.388	0.333	0.462
Question Labels w/ Resampling	0.653	0.354	0.454
Question Labels Binarized	0.212	0.303	0.439
Question Labels Binarized w/ Resampling (BayesNet)	0.634	0.188	0.472
Question Labels Ranked	0.614	0.215	0.459
PLDA	0.666	~	0.435

Table 3: LLDA Results

6.6 Results

Table 3 shows the overall results for the different models of LLDA, validated both using cross-validation and a single internal split of the training data (70% used to train, 15% used as a 'dev' set). All models were run with the better performing of NaiveBayes and BayesianLogisticRegression from Weka, with the exception of the one marked as the BayesNet. Note that the cross validation scores are just for the predictive power of the weighted training labels produced by the LLDA software, and do not account for the uncertainty of the labels produced by LLDA, but are effective for comparing the models to each other. These are the scores that the model would be expected to have, assuming that LLDA worked perfectly and produced the true labels. Thus, the scores from the training/dev split over the training data are a much better metric for the final results since they accounted for both the uncertainty of the LLDA prediction and the uncertainty of the final generative model. However, if we could not get a very high cross-validation score for the question model, there would be no point in attempting to predict the question responses with LLDA, so it was still a useful metric to determine which model to use.

Since the only version of the binarized model that caused any improvement was also the only model that successfully used a Bayes Net so it was difficult to determine how much it actually helped as a better representation of the data. We also tried an alternate of ranking the questions that LLDA determined would be high by order of their probability, but this did not cause any significant increases.

6.7 Partially Labeled Dirichlet Allocation

Stanford's LLDA implementation (Ramage et al., 2009) allows customization, through use of a technique they call Partially Labeled Dirichlet Allocation. This facilitates customization of the number of topics per label and latent topics within the label inference (Ramage et al., 2011). We determined which labels needed additional topics manually (by analyzing complexity of the questions and quality of the results), and through programmatic feature-selection. Since the results of both techniques were nearly identical (manual selection was slightly better), for the rest of the discussion we will refer to them interchangeably.

Question 3:
I did not feel like eating; my appetite was poor
het nie wat tak jak mijn che een maar die met dit dat har weer ale nem dag net nog
Question 6:
I felt depressed
music find thing found long watch year show dream thinking facebook week book didn computer feeling morning years doesn making

Table 4: Least Predictive LLDA topics

Among the least predictive questions in LLDA analysis were Question 3 and Question 6, as shown in Table 4 The latter makes sense, since it is as difficult an answer to predict as the final class of the model. Although only 6 of the 20 top topics were the same as the overall depression prediction. Question 3 seems like it would be easy to predict by watching for discussion of food or eating, yet its topics came out as a set of nonsensical foreign words.

To attempt to correct these poorly performing labels, they were given either 1 or 2 extra topics depending on their performance. Additionally, we created a latent state in an attempt to emulate the underlying overall classification. We then ran LLDA inference as before and plugged the results into our various predictive models in Weka, ex-

***Latent*:** week weekend tonight finally morning hours year tired class birthday yay snow long bed car fun facebook wait nice phone
Q6 (0): movie answer dream watch watching show year minutes game points question long hours find stupid read book half years play
Q6 (1): found find computer bed thinks thinking boys check pain sort made learnt children feeling bad post job facebook sick big
Q6 (2): che ako ang sono haha con naman mga kung hindi gli tutti perch din questo nel talaga poi cos sempre
Q3 (0): tak jak ale nem nie pro por para est los ten nen mas dias lov pak subject asi nov todos
Q3 (1): het wat mijn een maar die dat met dit weer nog nie net niet gaan naar kan heb aan lekker
Q3 (2): bored dont fuck hahaha haha facebook hate watch guy yeah yay bad fuckin wait head lolz stop big guess year

Table 5: Additional Topics Under PLDA

cept this time with 42 features per user to represent each of the new topics.

As a result of this technique (Table 5), two of the new topics for question 3 continued to be gibberish, but it did successfully pull out a useful 3rd topic. Although it does not seem to refer to eating, as we expected, it does bear some resemblance to intuition about depression. Interestingly, however, the most predictive of these new topics was topic 1, which is very similar to its original topic list. Perhaps a useful technique that could be applied in future work would be to allow each question several hidden topics, and then select features from the ones that produce the most cohesive results.

Looking at the results of the chi-squared feature analysis in Table 6, we can clearly see that PLDA did indeed improve the usefulness of the features as a whole, since the highest and lowest scores all got higher. Surprisingly, topic 1 from question 3 moved to the top of the list, even though it was still nonsensical. This probably caused some overfitting of the data, as a rare foreign word was being incorporated in over all prediction, when it really could only identify a few specific users. A similar effect can be seen with question 13, also a foreign language topic. A specific example of the effects of PLDA in general is question 6, which jumped from the bottom to the upper half thanks to its having secondary topics which could partition away some of the more useless words for that question.

LLDA	
Score	**Attr**
119.0	q20
113.0	q10
106.5	q14
...	
81.4	q13
78.6	q11
71.3	q3
...	
42.3	q2
35.6	q7
28.7	q6
0	q17

PLDA	
Score	**Attr**
188.0	q10 (0)
174.9	q13 (0)
...	
121.1	q3 (1)
...	
120.8	q20 (0)
...	
100.6	q6 (0)
...	
80.0	q13 (1)
77.9	q3 (2)
69.2	q3 (0)
...	
53.7	q6 (2)
51.4	q6 (1)
48.5	q8 (0)
39.2	q2 (0)

Table 6: Chi-square Analysis with and without PLDA

Overall, PLDA appeared to be a valuable extension to LLDA for improving the predictive power. Although it increases the number of features and improves the accuracy for each label, it seems to suffer from over-fitting. In fact, since the simplest model of PLDA (0 latent topics and 1 topic per label) is equivalent to LLDA, it shouldn't do worse than regular LLDA except for because of over-fitting.

7 Results and Analysis

Our baseline from the last semester was a good start, giving an F score of 0.34 on the test data. This was certainly better than the numbers we got when we were predicting *neuroticism* in the last semester on a similar data set.

The attempt to have a flat part of speech tags did not help much, the reason being the data was too dirty and status messages are often malformed for the tool to successfully parse . Also, the F score on cross validation was poor. There may be a possibility to improve this method by having a more structured POS parse tree. Our attempts to apply deep learning were not very fruitful, however they did give us good insight into the data and we were able to learn the deep representation of

Models	**F-Score** (Kaggle and CV(Cross Validation)
Baseline: Bow + LDA +LIWC	0.32, 0.34
Bag of Tags + LDA + LIWC	0.32 on 10-cv
Deep learning : Flat tree + Parsed tree	0.24, 0.37 on 10-cv
sLDA: Just labels , Questions based	0.47, 0.64 on 10-cv
Active learning +llda	0.39, 0.803 on 10-cv
pLDA	0.38, 0.66 on 10-cv

Table 7: Summary of Results

words. One of the degrading factors was the inability to parse the text we get from facebook status messages. Also we found that it is extremely difficult to combine these vectors and make sense of the meaning as a whole.

Our attempt at active learning has provided us with an alternate approach which utilizes human intuition. Even though the prediction score was not high. We believe that there is a clear distinction between the depression related (**consistent negative signal**) statuses from the rest. One possible explanation is that in a short communication channel like Facebook, users want to expose as much information as possible. The best way to do so is to pick the highly informative words that reveal their emotion. Discarding depression unrelated statuses from the dataset may not help prediction because this removes information. However, this approach unlocks new potential direction to solve the depression prediction problem. Understanding the deeper theory of human languages used in the context of public communication may further help us create the right annotation tasks. Feeding the outputs from multiple annotation task into an ensemble approach may be an interesting idea we would like to explore in the future.

Our model of prediction in LLDA was the best performing model. We had succeeded with both single label and multiple label per document. We tried to extend LLDA by using PLDA. Initial results were decent, but did not appear to significantly improve our LLDA model, so we did not pursue the technique further. It is entirely possible that the right combination of hyper-parameters could yield good results, especially if the data is such that a tightly fitted model is more helpful, and our further work on the subject will explore methods for selecting these parameters. This requires further investment of time, and there is a possibility of improvement through this route.

8 Conclusion

Especially in comparison with our previous work in predicting neuroticism through facebook posts, we felt we had some good success in predicting depression, and see this as a potentially valuable tool in the field of clinical psychology. Although there is certainly room for improvement from our results, through further model refinement, we have demonstrated that there is an informative signal that can be extracted from the noisy data of Facebook posts, and have suggested some effective tools for extracting it.

References

[evin Gimpel and Smith2011] Brendan OâĂŹConnor Dipanjan Das Daniel Mills Jacob Eisenstein Michael Heilman Dani Yogatama Jeffrey Flanigan evin Gimpel, Nathan Schneider and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. ACL.

[James W. Pennebaker and Francis.2007] Roger J. Booth James W. Pennebaker and Martha E. Francis. 2007. Linguistic inquiry and word count. ACL.

[Mohit et al.2014] Iyyer Mohit, Enns Peter, Boyd-Graber Jordan, and Resnik. Philip. 2014. Political ideology detection using recursive neural networks. Association for Computational Linguistics, June.

[Ramage et al.2009] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256, Singapore, August. Association for Computational Linguistics.

[Ramage et al.2011] Daniel Ramage, Christopher D. Manning, and Susan Dumais. 2011. Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 457–465, New York, NY, USA. ACM.

[Resnik et al.2013] Philip Resnik, Anderson Garron, and Rebecca Resnik. 2013. Using topic modeling to improve prediction of neuroticism and depression in college students.