CS584: BIONLP Project 2

Ramon L. Correa
[1]**Emory University , Atlanta, Georgia , United States**

1 METHODS

We obtained a dataset consisting of 116 patients of individuals prone to suffering from falls. Patients were paired with their spouses to help monitor each other's falls and provide reports of the event. Each patients fall was classified into a Center of Mass (CoM) (n=81) , Base of support (BoS) (n=23) or other type of fall (n=12). For the purpose of this study we are only interested in COM falls vs all others. Our dataset was thus stratified onto 81 CoM patients Vs 35 other falls. Our Study we set out to build a machine learning based pipeline to process the patients text reports.

1.1 Building Text Features

Text reports by themselves are not informative to a program attempting to a classification. To obtain meaningful information we opted to generate a diverse set of features that would bring out potential contextual information that could thus be exploited by the classifier.

N-Gram extraction
One of the most features in NLP and information retrieval is the use of ngrams of lengths 1,2 as they allow us to obtain better context of what's occurring in the document. Text n-gram count can be found by using the sklearn countVectorizer object. Before the feature extraction is done the program pre-process the text by normalizing cases by forcing text to be lower case. Stop words were not removed under the idea that their use may be of actual use.

Term Frequency Inverse Document Frequency
This text reports cover similar topic field. We suspected that similar falls may have subtle similar wordings which may be identified by having similar term frequencies.

Location Lenght information
Some of the fall events may be influenced by their locations. Such as being in high risk areas or because of activities in that area (such as being distracted while cooking). We noticed there was a variability in the description of the location. We suspect that some fall descriptions may have a location description with more text as the patient tries to be more precise if it's a situation of concern. Meanwhile less serious falls may not have such detailed descriptions.

Text polairty
The reports consist of patients self report of the situation. Their recollection of the events may vary depending on the severity of the event and henceforth influence the text. Using NLTK's pre-trained text polarity model we obtained the average polarity measure of the Text reports sentences. Each patients report thus had an average measure of negativity, neutrality and positivity.

Text Subjectivity
Following a similar line of thought to the previous segment more involved situations regarding falls will have more personal details we hope to pick up using this metric. This particular model was pre-trained using the subjectivity dataset from NLTK. Once trained the model could provide a single metric describing whether some text was ”objective” or ”subjective”. For this sample we decided to opt to use the binary classification instead of using confidence scores.

1.2 Experiment 1: Identifying useful classifiers

Once we established the features for our model we needed to build a model selection methodology. We did this by using sklearns grid search with cross validation. We had 6 different candidate models with a series of potential parameters that needed to be tuned. The grid search class in sklearn allows us to easily run multiple permutations and evaluate the dataset. In table 1 bellow we see the hyper parameters that were tuned for each one of them. The Models were optimized with respect to the f1-micro score.

Model	Parameters
Tree	`Criterion(Gini,entropy),max_depth(10,30)`
Random Forest	`n_estimators(5-20),max_depth(4-8)`
Naive Bayes	`fit_prior(true,false),alpha(0-1)`
SVM	C(0.2-20)
Logistic Regression	C(0.2-20)
Quadratic Discriminant	regularization(0-1)

Table 1: Series of models trained alongside the ranges of the hyper parameters the model was tuned for.

1.3 Experiment 2: Identifying relevant feature set

Once Model selection was completed we obteed to select the top performing models and measured how their performance was altered by varying the feature set being extracted. We ran different combinations of the feature set with a locked down classifier to observe which feature set was most performative. Based on the best performing classifier and feature set combination we would then move on to the final experiment

1.4 Experiment 3: Training size influence

The final test would be to observe how much the model is influenced by the size of the training set. Models such as random Forest and SVM have been known to be dependant on the training set size. We will therefore simulate varying datasets by having the training set be varying percentages of the entire dataset starting from a 10-90 split onto an 90-10 split.

2 Results

2.1 Useful Classifiers

In table 2 we observe random forest has the largest average performance. Our baseline model was the naive bayes model utilizing the same features

Model	Average F1$_micro$
Random Forest	.7934
SVM	.7416
Logistic Regression	0.7074
Decision Tree	.7065
Naive Bayes	.68012

Table 2: Average model performance ranked from best to worst

2.2 Relevant feature set

Num features	Accuracy	F1$_micro$	F1 macro	Features
2	0.7185	0.7185	0.5491	Counts,Polarity
3	0.7361	0.7361	0.6409	Counts,polarity,subjectivity
4	0.7010	0.7010	0.5765	Counts,tfidf,polarity,subjectivity

Table 3: Caption

Imputation Study

Once the random forest model was finalized we varied our test set size to be x percentage of our dataset size. The impact on performance on each metric is reported in Figure 1.

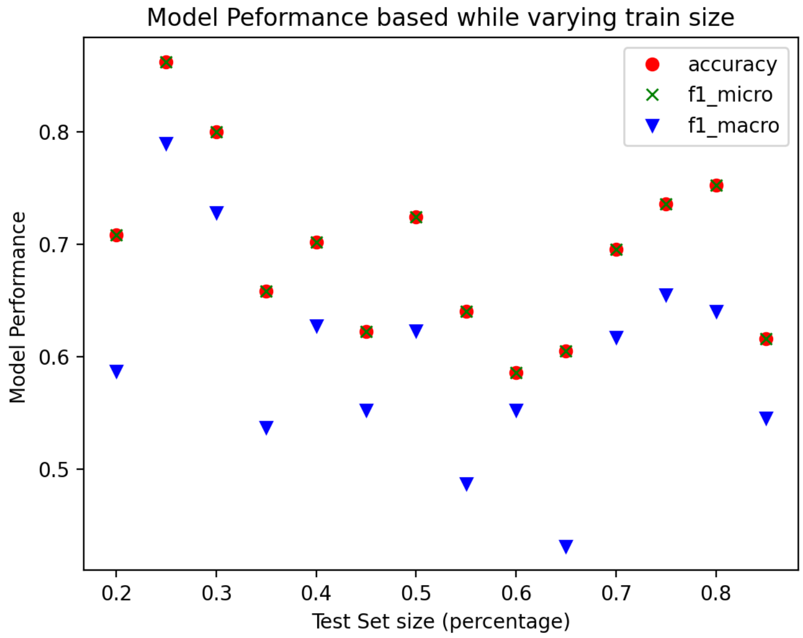

Figure 1: Caption

Discussion

Out of the complex features generated we identified Counts, polarity and subjectivity were the best performing features. Our classification accuracy reached the 86% range for the small test set size. For test set sizes of 50% or more we perform slightly worse than a classifier that would have predicted the majority. The fact polarity and subjectivity were still identified to be relevant features suggest there may be some contextual information they are picking up. It would be worthwhile to investigate the usefulness of these features.