2014 CS229 Stanford University

r/Classifier

Team Members

Andrew Giel, Jon NeCamp, Hussain Kader

Our Task

Given a text reddit post composed of a title and body, we classify the subreddit the post belongs to. This can serve two main functions:

- to lower the barrier to entry for new users to Reddit who do not know which subreddit to post to,
- to suggest which subreddit a post will be most successful in, helping users to achieve high visibility for their content.

Figure 1: Our project classifies reddit posts into their respective subreddit

In order to make this problem tractable, we have limited the scope of reddit posts to only text posts. Thus we have defined a text classification problem.

Data

Our dataset can be found at:
https://github.com/umbrae/reddit-top-2.5-million
We are using data from twelve subreddits:

NoStupidQuestions	*shortscarystories*
Showerthoughts	*DebateReligion*
confession	*relationship_advice*
UnsentLetters	*self*
askphilosophy	*ShittyPoetry*
AskMen	*AskWomen*

These subreddits were chosen due to their high number of text posts and lack of distinctive features that would trivialize the task (eg. 'TIL').

Features

Title Splitting

Our knowledge of reddit gave us insight into what we found to be a very powerful feature, the title and body splitting.

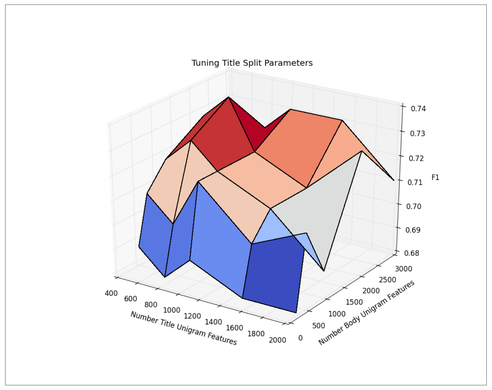

Figure 2: Tuning Title Split Parameters

Mutual Information

We selected the top $n = 3000$ features using the notion of Mutual Information (MI), defined as follows

$$MI(C, w_i) = \sum_{k=1}^{|C|} P(c_k, w_i) \log \frac{P(c_k, w_i)}{P(c_k)P(w_i)} + \sum_{k=1}^{|C|} P(c_k, \bar{w}_i) \log \frac{P(c_k, \bar{w}_i)}{P(c_k)P(\bar{w}_i)}$$

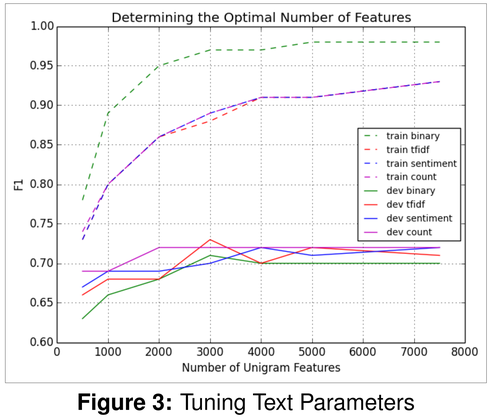

Figure 3: Tuning Text Parameters

TF-IDF

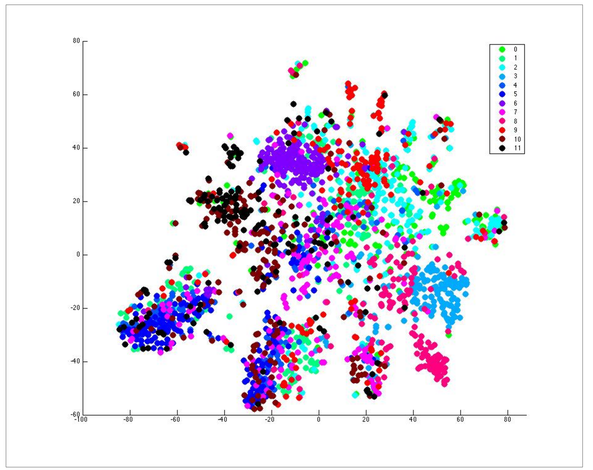

Figure 4: t-SNE visualization of our tf-idf feature space

Models

We primarily used two models: Multinomial Naive Bayes and Multinomial Logistic Regression.
Multinomial Naive Bayes is a generative model that learns $p(c_k)$ and $p(x_i|c_k)$ during training. Given some new input x to evaluate, it predicts the class via

$$\arg\max_c p(c) \prod_{i=1}^{n} p(x_i|c)$$

Multinomial Logistic Regression trains via Stochastic Gradient Descent, learning some parameters θ to minimize the cost function $J(\theta)$

$$J(\theta) = \frac{1}{m}\left(\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2\right) + \frac{1}{C}\sum_{i=1}^{n}\theta_i^2$$

where

$$h_\theta(x^{(i)})_j = \frac{\exp(\theta_j^T x)}{\sum_{k=1}^{K}\exp(\theta_k^T x}$$

which is the softmax function.

Results

Model	P	R	F1
Naive Bayes	.70	.68	.68
Logistic Regression	.75	.75	.75

Considering this is classification over 12 subreddits we are very happy with this outcome. Below are the subreddit by subreddit results for our best model (Logisitic Regression using Title Split, tf-idf and a word count feature)

subreddit	P	R	F1
NoStupidQuestions	.66	.90	.76
shortscarystories	.80	.73	.76
Showerthoughts	.84	.83	.83
DebateReligion	.89	.91	.90
confession	.77	.78	.78
relationship_advice	.74	.72	.73
UnsentLetters	.84	.85	.85
self	.68	.67	.67
askphilosophy	.89	.73	.80
ShittyPoetry	.80	.83	.81
AskMen	.57	.52	.54
AskWomen	.59	.57	.58

Regularization

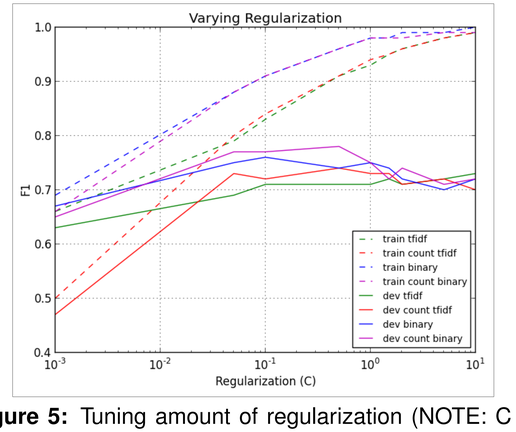

Figure 5: Tuning amount of regularization (NOTE: C is inverse regularization)