SENTIMENT CLASSIFICATION ON STEAM REVIEWS

Stanford | ENGINEERING
Computer Science

Rohan Bais (rbais@stanford.edu) | Pascal Odek (pascalo@stanford.edu) | Seyla Ou (seylaou@stanford.edu)

Predicting

Steam is a multi-billion dollar distributed gaming platform with a young user-base (18-30). The culture around Steam is built on sarcasm, memes and wit in addition to it being more hip and modern-lingo based. This makes binary sentiment classification on reviews difficult. We built 5 models to classify the sentiments on steam reviews to see which models would overcome the nuances presented on this platform. The best performance was by SVM > Logistic Regression > Naive Bayes > Turney's Algorithm > Lexicon Baseline.

Data

The data was scraped off from Steam's game pages using custom Javascript HTML parsing. Since we wrote the code to extract data, we extracted data relevant to our features: the text, hours played, and percent who found the review funny and helpful.

Features

Available: # hours played, % found useful, % found funny

Derived: TF-IDF + positional weighting

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Models

Lexicon Base: Compare # positive/negative words

Naive Bayes: Predict class based on bag of words

SVM: Hyperplane classifier using TF-IDF features

Logistic Regression: Similar usage to SVM

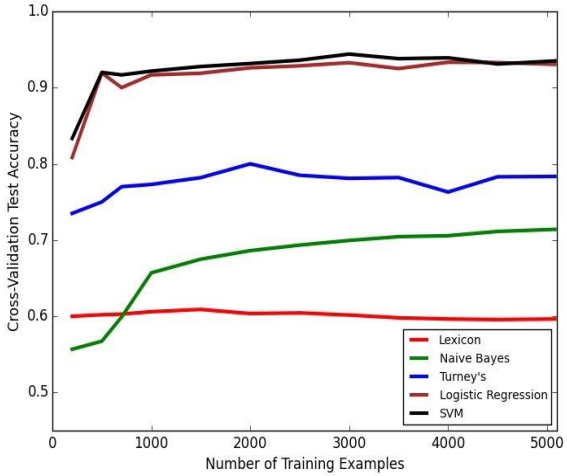
Turney's: Find phrases, determine semantic score using PMI, average score

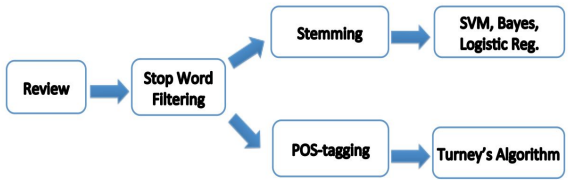
$$\mathrm{PMI}(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$

Testing and Results

Done with a 10-fold cross validation of 70/30 percent training/test split, which is approximately 3500 reviews and 1500 reviews respectively for each fold.

Method	Train	Test	F1
Lexicon Aggr	59.6%	59.6%	66.1%
Naive Bayes	86.8%	71.4%	74.8%
Logistic Regr.	93.5%	93.1%	94.4%
Linear SVM	93.9%	93.5%	94.7%
Turney's Alg.	79.2%	77.6%	73.0%





Discussion

- The Baseline and Naive Bayes approach suffered the same problem of losing context of words and not preserving the order of the sentiment.
- Turney's Algorithm suffered less from this issue, but still tripped up in context (e.g. found a phrase that would describe a plot or premise of a game that would involve negative phrases, but sentiment would still be positive). Drastically does better with access to huge datasets to properly determine semantic orientation of phrases.
- Logistic regression and Linear SVM outperformed both, as expected, due to details such as TF-IDF correctly weighing the indicative words and hours played and other features being good detectors of sentiment outside of the words in the review (e.g. many hours played means most likely positive sentiment).

Future

For future work, we were considering trying out more features (e.g. if product was a gift, word2vec embeddings) and perhaps trying out CNNs or RNNs to determine how neural networks fare relative to these methods. Also, much of the groundwork for sarcasm detection is laid out here, so another interesting problem to tackle, using the same or additional features, is to detect sarcasm in text.

References

Ramos, Juan. "Using TF-IDF to Determine Word Relevance in Document Queries ."Rutgers.edu, 10 Jan. 1999,citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1424&rep=rep1&type=pdf.

Lee, Lillian, et al. "Thumbs up? Sentiment Classification Using Machine Learning Techniques." Cornell CS Journal.