Predicting Popularity of Fanfiction Stories Based on Title and Summary

Aojia Zhao
aojia93@stanford.edu

Abstract

Concisely and quickly conveying key information is crucial in online media interaction with users today. Most importantly is being able to grab readers attention with buzz words and popular phrases. Fanfiction sites serve as a prominent example of the importance of short summaries influencing potential readership and popularity of fan written stories. With millions of available stories to sift through, it is crucial for authors to create attractive titles and summaries. In this project, a variety of machine learning techniques are used to help automatically predict future attractiveness of short summarizations. Baseline classifiers include LogisticRegression, StochasticGradientDescent, and Support Vector Machine. Neural network architectures include Recursive Neural Network. The resulting classification holds good promise with 72.9% accuracy.

1 Introduction

In today's fast paced world of online information exchange, the ability to engage and communicate with users efficiently is a must for any media outlet. This is even more important for traditional media like text, where a simple title or summary can make or break view counts and popularity ratings. Previously, efforts have mostly been in summarizing and evaluation of main document corpus rather than short abstracts and titles.[1][2] These efforts, although exhaustive, tend to be inefficient and loses sight of first impressions presented to readers through summaries. In this project, a variety of investigative techniques and algorithms are applied to help automatically predict how attractive and interesting given titles and summaries are.


Harry Crow > by robst *reviews*
What will happen when a goblin-raised Harry arrives at Hogwarts. A Harry who has received training, already knows the prophecy and has no scar. With the backing of the goblin nation and Hogwarts herself. Complete.
Rated: T - English - Chapters: 106 - Words: 737,006 - Reviews: 25502 - Favs: 15,251 - Follows: 12,044 - Updated: Jun 8, 2014 - Published: Jun 5, 2012 - [Harry P., Hermione G.] - Complete

Dataset for this project will come from Fanfiction.net, a popular hub hosting the largest repository of freely available fan written content currently. As a generalized overview, above is a sample story entry with key features like title, summary, word count, and review count. In this paper, the focus is on classification of story popularity based on its title and summary. Review count is used as the output metric as in general it is widely accepted in the community as a good metric of popular stories.

The main plan of attack includes using traditional techniques like Logistic Regression and Support Vector Machine as a control performance evaluation and comparison metric, while a recursive neural network model will be used to test the hypothesis that deep learning approaches can produce better generalized predictions as opposed to fixed dimensional inputs of human labelled features. The baseline classifiers will be implemented using Numpy and Scikit running on default hyperparameters while the neural network architecture will be based on the Tensorflow tutorial code available at www.tensorflow.org.

2 Background/Related Work

Prior literature on this topic mostly focused on clickbait title detection, which is usually viewed in a negative light. [3][4] Clickbait titles are sensationalist titles designed to attract readers to read the article, without promise of objectively truthful content. In these papers, a variety of news articles from NYTimes, Royal Guardians, Daily Mail, etc. were used with their professional writing qualities held in mind as high level to limit need for syntactical preprocessing. Approximately 5500 labelled data pairs were used to train and test, with 3000 neutrals and 2500 clickbaits. Using k-nearest neighbors as a baseline, certain features including flashy vocabulary and famous celebrity names were found to correlate closely with high reader retention rate. When pronoun and proper nouns were processed out, accuracy dropped drastically, suggesting them as key pivot words in holding reader focus. In a follow up classification using SVM with bag-of-words approach, training on single topic domain reached 97% accuracy, while cross-domain classification dropped significantly to 75% accuracy. This suggests different genres have different key features associated with potential popularity and a traditional hand labelled dataset can't be generalized very well.

This idea is further investigated in the deep learning approach on cross-domain sentiment analysis paper by Glorot, Bordes, and Bengio. [5] In this paper, a stacked auto-encoder model took in sentence level word vectors as input and preprocessed off irregular vocab to denoise the data. The hypothesis is that with multiple layers inherent in deep learning systems, the strongly hierarchical structures in sentence phrases can be learned more generally. Using a dataset of 340,000 Amazon reviews over 22 topic domains, the paper evaluates success on a transfer error ratio defined to be variation of individual output over the mean output variation. A baseline classifier of linear SVM was able to achieve an error of 14.5% while the full neural network system with 3 hidden layers and 5000 hyperbolic tangent units achieved a significant improvement of 10.9% error. This result outperforms that of single domain classification state-of-the-art models, which produce around 13.9% error. While only vanilla feedforward neural networks were used in this paper, the general applicability of deep learning approaches to categorical popularity classification is without doubt.

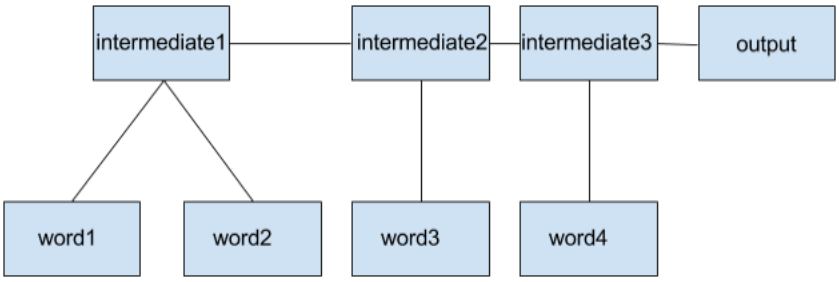
3 Problem Statement

The main problem statement is the following:

Given a sequence of word tokens $t_1, t_2, ..., t_n$ representing summary/title, create a classification system to determine the potential range of review count the story will receive when published. To convert from a continuous output over all positive integers from 0 to 35,000 (the upperlimit on most reviews of a single story in the dataset), output will be evaluated in three bucket ranges of accuracy: 0-10000, 10001-20000, and 20001-35000. If the output raw review count belongs to the correct bucket, classification will be considered correct. As there are 3 output classes available, the output will be a 3x1 vector denoting probability of review count range bucket.

4 Technical Approach and Models

The main model will be a Recursive Neural Network model that maps input word vectors to outputting the probability of each of three potential bucket classes. Given a sequence of word tokens, the system first converts each word to corresponding 50 dimensional word vectors using word2vec. During this process, preprocessing eliminates common stop words and irregular vocab words present due to misuse, misspelling, etc. For words without a corresponding word vector, the unknown token word vector is used instead as a fallback. After word vectors are processed, the sequnce is read in order, combining two consecutive word vectors together to output an intermediate vector, which is passed on to the next vector in the sequence. This is demonstrated in the figure below.



The general probability of each bucket can be expressed as:

$$\begin{aligned} P(y|t_1, t_2, ..., t_n) &= \hat{y} \\ &= softmax(Uh^{(n)} + b^{(s)}) \\ h^{(n)} &= sigmoid(Wh^{(n-1)} + t_n + b^{(1)}) \end{aligned}$$

The dimensions are

$$\begin{aligned} h^{(n)} &\in R^{50x1} \\ \hat{y} &\in R^{3x1} \end{aligned}$$

As in previous assignments, the cross entropy loss is used to find gradients to train the network.

$$CE(y, \hat{y}) = -\sum_i y_i log(\hat{y}_i)$$

As summaries generally tend to be around 30-50 words, the danger of vanishing gradient can be a real concern. In order to avoid this scenario, every 5 words are reconsidered as one individual vector entry into the system. For example, given the sentence "What will happen when a goblin raised Harry arrives at Hogwarts," the first five words "what will happen when a" will go through the entire system to produce a cumulative word vector, which is then considered constant for the rest of the network and not used for chain rule in gradient calculation. This is similar to context window fixing for CBOW done previously but applied linearly without repeat. The expectation is this strategy should diminish effects of vanishing gradients on multiple levels of recursive model.

5 Experiment

5.1 Data Gathering

Data was gathered primarily from the Harry Potter section of Fanfiction.net using a python script to parse the HTML source code of pages containing stories ranked by review count. In total 47075 entries were gathered, with review counts ranging from 34956 to 45. Due to the nature of only a small percentile of stories holding extremely high review counts, the data points were separated into the three respective buckets. In each individual bucket, 10-fold cross validation was used to generate 90% training data and 10% testing data. In order to tune hyperparameters, a further 10% of training data was used for development and excluded from training. In total, the data distribution was as follows: testing-4708, developement-4708, training-37659.

5.2 Baseline Classifier

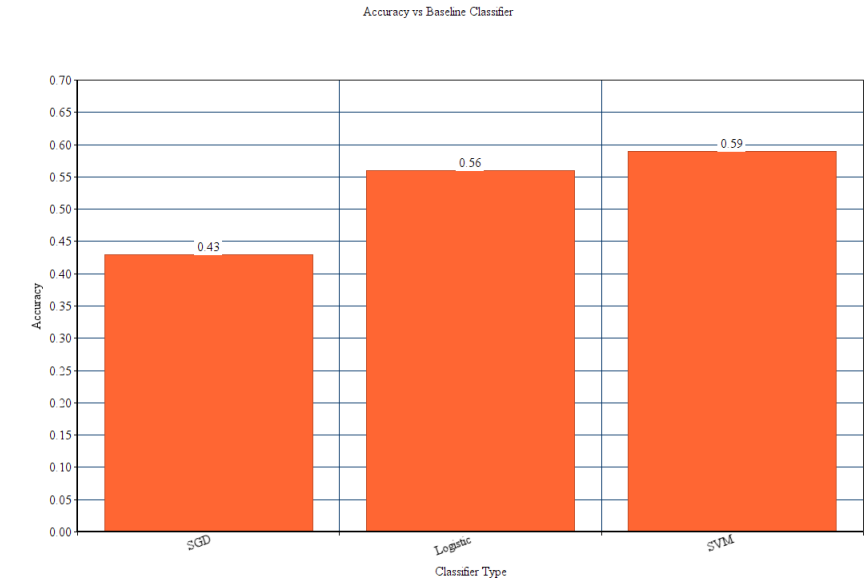
For the baseline classifiers, Logistic Regression, Stochastic Gradient Descent, and Support Vector Machine were used to predict categories of summaries and titles. Because these models dictate fixed dimensional sizes for input features, the word vectors were preprocessed like in the case of RNN, but subsequently averaged together to produce a "summary" word vector. The hypothesis is this should reduce the influence of outlier vocab token's influence while distilling the core idea into the 50-dimensions. For implementation, Scikit built in library functions were used for all three, with default hyperparameters of 0.001 learning rate and 1000 iterations for training.

5.3 Evaluation Metric

Because this is a three category classification task, quantitative analysis will be on percentage of correctly classified outputs. The following section presents results from baseline as well as RNN models.

6 Results

6.1 Baseline



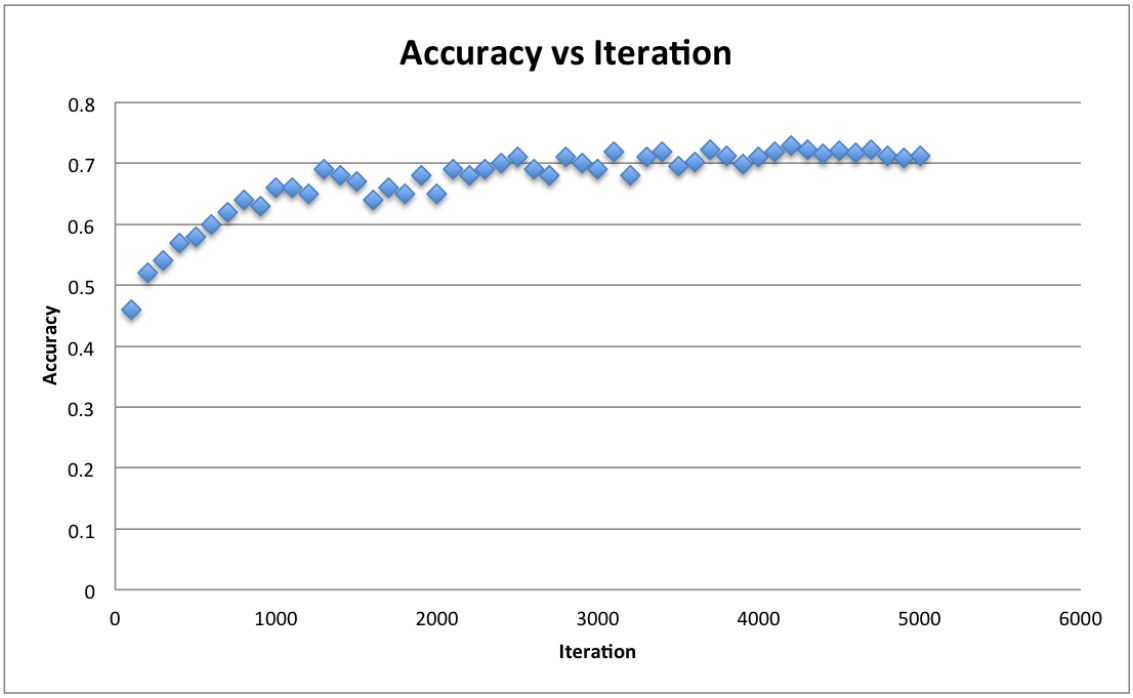
The baseline classifiers show relatively expected results of performance between 40% and 60% accuracy. In order, Stochastic Gradient Descent performed the worst, only better than randomly guessing classification by 10%. Performing much better are Logistic Regression and SVM. One explanation for this is that LR and SVM are much more suited to classification tasks while SGD serves better in general continuous domain of outputs. This is especially true in this case, where SVM achieved the best result as it theoretically defines a boundary separating individual categories. A potential reason for the poor performance of all three baseline classifiers is the averaging process which accumulated all input word vectors into one. Although this serves to denoise input word vectors, the influence of specific pivot words are also lost as no weighting is done to emphasize specific vocabs. Take the following story summary as an example:

One lonely little boy. One snarky, grumpy git. When the safety of one was entrusted to the other, everyone knew this was not going to turn out well... Or was it? AU, sequel to "Harry's First Detention". OVER FIVE MILLION HITS!

Gold Label: (10001-20000) SGD: (0-10000) Logistic: (0-10000) SVM: (10001-20000)

The subtle pivot words here, to a human reader, are the "Or was it?" and "OVER FIVE MILLION HITS!". These short phrases can poke at the strong emotional feelings of readers to want to know why there's a twist and why it has over 5 million hits. However, with averaging word vectors, the effects of these key vocab tokens are diminished and thus incorrectly classified.

6.2 RNN



Training the RNN until different iterations and evaluating accuracy results in the above graph. Each data point is spaced by 100 iteration intervals. It is clear that the initial rate of improvement is very high, as by iteration 1000, the accuracy has already reached 0.671. From that point on, the accuracy steadily improves until plateauing around 0.720. The maximum reached was 0.729 at iteration 4200. Subsequent iterations after 5000 were not continued as the accuracy appears to have stabilized. Overall, this performance is very encouraging as the baseline classifiers were not able to break even 60%, suggesting the individual word vector inputting has drastic effects compared to averaging them all. Furthermore, the 5 word accumulation strategy appears to have halted the vanishing gradient problem, though around iteration 4500, some semblance of decline in accuracy is starting to appear.

An example of the effects of specific words being considered is in the correct classification of the following story:

Summer before third year Harry has a life changing experience, and a close encounter with a dementor ends with him absorbing the horcrux within him. Features Harry with a backbone.

Gold Label: (10001-20000) RNN: (10001-20000)

Although very short in length, the RNN model was able to find the pivot words influences of strong tokens like ”backbone” and ”life changing”. This is different from the baseline classifiers, which all failed to classify this entry correctly.

7 Conclusion

Overall, this project has shown potential applicability of RNN in predicting the popularity of fanfiction story titles and summaries using review count as an evaluation metric. Using baseline classifiers, SVM achieved an accuracy of 59%. For the main architecture, the neural network was able to reach 72.9% accuracy in under 5000 iterations. With this said, there are many future investigations possible to extend upon the results.

Most importantly, due to the time constraints in training and evaluating data, the sample dataset was only able to be surveyed over one domain of stories, Harry Potter. Although general language

elements tend to remain the same across story genres, It will be useful to expand to other genres of text summary and titles. As this model is generalized to all texts, data points beyond fanfiction can also be used to investigate differences between text sources like news articles, blog posts, etc.

Another potential path is in incorporating more features of evaluation beyond just review counts and summary/title word vectors. One easy extension is to incorporate actual text body of reviews left by readers and do a sentiment analysis. This could serve as an alternative source of evaluation for popularity prediction. Other available elements parsed from the story set but not utilized include word count, favorite count, etc.

8 References

[1] Cho, Kyunghyun, Aaron Courvilee, and Yoshua Bengio. "Describing Multimedia Content Using Attention-based EncoderDecoder Networks." Cornell University Library. ArXiv.og, 4 July 2015

[2] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A Neural Attention Model for Sentence Summarization." ACLWeb. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing

[3] Lex, Elisabeth, Andreas Juffinger, and Michael Granitzer. "Objectivity Classication in Online Media." ResearchGate. Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, 13 June 2010.

[4] Chen, Yimin, Niall J. Conroy, and Victoria L. Rubin. "Misleading Online Content: Recognizing Clickbait as "False News"" ResearchGate. ACM WMDD, 9 Nov. 2015.

[5] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "Domain Adaptation for Large-Scale Sentiment Classi Cation: A Deep Learning Approach." WUSTL. Proceedings of the 28 Th International Con- Ference on Machine Learning, 2011.