Automatic classification for Fanfiction Websites: A Solution To Mistagging Due To Human Error

Samuel Kamenetz Ithaca College
skamene1@ithaca.edu

Abstract

Using machine learning, corpuses of fanfiction can be automatically classified, an alternative to user assigned genre tags. This paper examines stories on the website fimfiction.net, and finds that by using Support Vector Classifiers for multilabel classification, we are able to achieve a better than random baseline.

I. Introduction

Fanfiction websites provide a community for users to read and submit pieces of writing about their favorite works of fiction. The typical structure of a site allows users to post their stories, and assign tags from a set list. The tags serve the important purpose of better defining the content of the story, and to facilitate searches by the readers. Likewise, there is always some sort of audience feedback mechanism, both qualitative (eg comments or reviews) and quantitative (view counts, likes, follows etc). Tagging is a powerful system for allowing users to find the content they want, but the effectiveness is mitigated if the tags are inaccurate.

In my personal experience, I've read many fan fiction stories that the author chose tags that did not match the story. There are often many reasons for a story to mistagged. Often the author, a novice writer, is unfamiliar with the site's common set of tags, and how they tend to be used. Perhaps the author did not understand their own story very well, thinking that the story was part of one genre, when the audience has a vastly different interpretation. Many authors, in attempts to get more traffic to their work, will select as many tags as they can, even inaccurate ones. Sometimes they mean a story to go a certain direction, but have not yet reached that point.

Some sites have different tagging rules to ensure better accuracy, to mixed results. Archive Of Our Own (known as AO3), is a notable exception to the traditional model. The website uses a folksonomy for their tags. They have free form tags, enterable by text box. While this allows greater specificity, there is no ranking of the relative importance of each tag, and no way to vote on which ones are accurate. The way that tags are created and applied can be esoteric, and miss the overall message of (eg, a story may be tagged with 'cuddling', 'first kiss', 'eventual romance', but not 'romance') While the tags on AO3 may be rich in providing information, the variety and specificity of the tags obfuscate what are the key features.

Fanfiction.net, perhaps the most well known fanfiction site, limits its users to having two genre tags assigned to their story; they are universal to all fandoms. One of said categories is 'general'. The limiting of tags may also fail to sufficiently describe the stories.

We seek an alternative to these systems. Our goal is to evaluate if document classification can be used to predict the genres of a work of fanfiction, and if so, to see if quantitative feedback mechanisms of the website can be used to distinguish between poorly tagged stories and properly tagged stories. This could serve as a basis for automatically ensuring tag accuracy.

II. Related Work

The standard model of fanfiction websites allows for any number of tags to be assigned to a given story. the nature of the tag structure calls for a multi-label classification for our study. This is a well studied structure in the field of document classification [2]. However, there are almost no examples of using text classification for corpuses of fictional works. The more well known corpuses used tend to be movie reviews, newspaper articles, and Tweets. Those that do look at fictional works are not concerned with the genre so much as being able to classify the authorship [3].

III. Proposed Solution

Our proposed solution is to test how well an automatic classifier can tag stories. Said classifier will analyze a story as a bag of words model, and assign what it predicts to be the appropriate tags. I will also determine what genre tags the system is better able to predict. The proposed metric for evaluation will be AUC, or Area under Receiving Operator Characteristic curve.

The Receiving Operator Characteristic plots the true positive rate (correctly identifying the presence of a tag, or recall), against the false positive rate (how often the system predicts the tag incorrectly, a measure of precision). each true positive moves the point on the graph up by $1/n$ (n being number of samples), and each false positive moves the point right by $1/n$. In the context of classification, the Area under the ROC is the likelihood that the classifier will assign a tag to a positive example, rather than a negative example. An AUC of 1 indicates perfect classification. A score of .5 indicates that the classifier is no better than randomly assigning a tag. [4]

I will also be testing a hypothesis that poorly tagged stories are viewed less, and this could be a predictor of what stories could be mistagged.

IV. Experimental Setup

I. Acquiring the Data

I chose fimfiction.net, a fan fiction website dedicated to the My Little Pony television series. I chose this because the site has robust metadata for each story, including likes, dislikes, comments, and views. The site currently uses 18 genre tags, and allow authors to select as many as they see fit. Second, in order to limit scope, I chose a site dedicated to a particular series and language, as opposed to fanfiction.net, which is encompassing of all series across all media, in all languages. Thirdly, a user of the site prepared an archive, with every story on the site as an EPUB file, and a JSON file of all the metadata. This was available for download as a 5 GB zipfile containing the vase majority of My Little Pony fan fiction in existence.

I parsed through the metadata, converting the EPUB files to something more amenable to Sci-Kit Learn's libraries. I wrote a parser that recorded the metadata in a SQLlite database, and converted EPUB to TXT files.

the stories trained and tested were selected randomly, choosing 14000 to train the classifier, and 1000 to test on. the test set was further divided into the third most viewed, and third least viewed, for the purposes of the second experiment.

II. Text Processing

From then on I used Sci-Kit Learn, a robust python library for machine learning. When processing text data, the first step is usually to represent each document as a n-dimensional vector, where n is the size of the vocabulary of the entire corpus (corpus being a collection of documents); each position represents a word in the vocabulary, and the value of the entry in that position represents how many times that word appears in a given document.

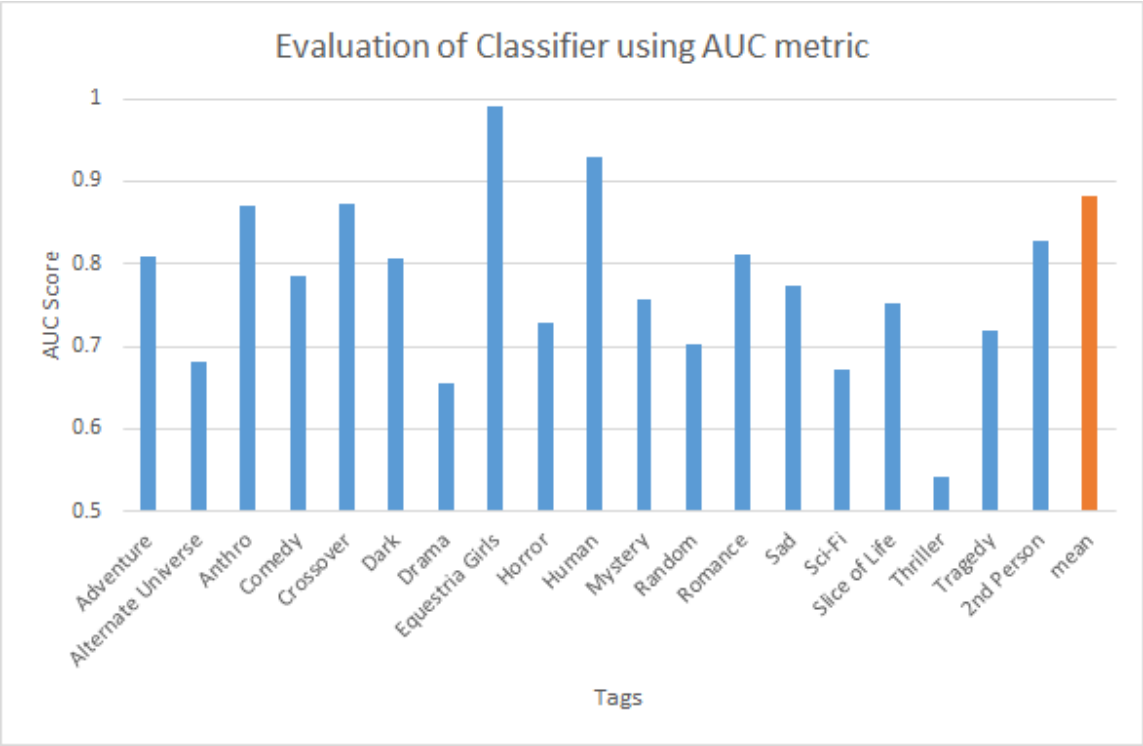
Once we have this vector representation of every document, I apply a tf-idf transformation to them all. It stands for term-frequency inverse document frequency. Bag of words vector model weights each word in a document by

the raw count, ie the count of how many times a given word appears in a document. For tasks like classification, this creates a bias towards longer documents, by sheer number of words. This essentially weights commons words lower when it comes to classification. The more documents a term appears in, the less evidence it contributes to belonging to a certain class. It also weights a certain term more heavily the more it appears in a given document. This prevents documents that are long from being more heavily weighted by the virtue of having higher raw word counts.

The classifier used is a linear support vector classifier, or linear SVM. Representing documents as n-dimensional data points, it constructs a hyperplane partitioning two classes, so that the margin between the hyperplane and the nearest data point is maximized. In our case, the two classes are "tag belongs", and "tag does not belong". When predicting, it determines which side of the hyperplane that the test data point falls on, yielding a positive or negative prediction, represented by a number betwen -1 and $+1$. the larger the magnitude of the numerical result, the more confident the prediction [1].

The classification problem in itself is a multilabel classification problem, Meaning a document can belong to any number of classes, as opposed to only belonging to one. The process involved is using a One Vs Rest classifier, which is actually a series of binary classifiers, one for each tag. Each is responsible for classifying the document as belonging to the tag or not belonging to the tag.

The resulting predictions for each document are compared with their actual tags, and then used as a basis for an AUC score to evaluate the overall effectiveness of the classifier.



V. Results

I. First Experiment

Using AUC as a metric, we find that in almost all cases, the classifier performs better than random in almost all genres (with the notable exception of 'Thriller', a relatively new tag. The 'Equestria Girls' was the most easily predicted, and 'Human' was a close second. 'Alternate Universe', 'Drama', 'Sci-Fi', and 'Random', performed relatively poorly.

II. Second Experiment

Table 1: *average AUC scores for each test set*

training set	AUC
entire test set	0.8815
top 1/3 most viewed of test set	0.9052
top 1/3 least viewed of test set	0.8527

There was a noticeable but small difference when classifying the least viewed of the test set versus the most viewed stories of the test set, a 0.05 difference. The largest discrepancy within a specific tag was under '2nd person', where the most-viewed test set received a score of .99, while the least-viewed test set was 0.62.

VI. Discussion

I. conjecture of particular patterns

The two most easily predicted genre tags were 'Equestria Girls' and 'Human'. I believe that these anomalies are due to the nature of the tags lead them to reuse a common set of words enough where those words become reliable predictors for that tag. For instance, the 'Human' as a tag, refer to human characters within the story (all characters in the show are either ponies or mythical creatures). Words that are normally absent from *My Little Pony* fanfiction, such as *hands*, *human*, and *feet* are very much present in works with the 'Human' tag.

the 'Second Person' tag also can be explained by the notion refers to a style of narration where the reader is directly addressed, usually as a character within the story, eg 'You open the door, and rub your eyes.'. It seems that words like *you*, *your*, and *you're* would contribute significant evidence to the second person tag. As for the discrepancy between the test sets, I believe that this tag is the most 'literary', and hence more likely to be misapplied by amaeteur authors. a second person story by a novice may not know the term, and not apply it, or they might apply it when it fits only a part of the story or none of it.

'Equestria Girls', likewise is a tag that denotes a story relating to the tie in movie of the same name. There are several characters that only exist within the canon of this particular movie, and the associated content. it also takes place in a setting largely foreign to the rest of the franchise, a mirror universe where all of the characters are human high-schoolers at Canterlot High. Between the established set of otherwise unique characters and otherwise unique setting, there are likely enough rare lexemes to accurately tag for this genre.

Unsurprisingly 'Random' had a low AUC score, .69 should you have any illusions about what this genre entails, here is this description from the website, "Stories that are very random and unpredictable, generally involving a lot of unexplained occurrences and behavior."

A central part of my thesis was that many works have improper tags, and a mistagged story will not receive as many views. Hence, few views would indicate a mistagged story. I was able to show that sorting tested stories by views had a marginal impact on performance of the classifier, though this is only a contributing factor. There are some latent factors that indicate mistagged stories I have yet to discover, if they exist at all, (besides having a human read the story). It may not be feasible to have an emperical method of testing whether or not the original tags are accurate. However, looking briefly at some examples from the testing set seem to confirm that even stories where the classifier did not predict all of the tags, it is sometimes easier to believe the classifier over the actual author's tags:

```
title:  Luna the Matchmaker
Description:
Princess Luna plays matchmaker for
the ponies holding the Elements of
Harmony.
predicted:
Romance, Slice_of_Life
actual:
Dark, Romance, Sad, Slice_of_Life
```

II. conclusions

In the future, I would like to expand on this project further, testing more classifiers, and under different parameters, perhaps even going beyond the bag of words model. With enough testing and fine tuning, automatic classification may become a feasible method for ensuring accurate tags. I would also increase the working size of my corpus. For time and space constraints, I was only to parse and categorize 15,000 works. there are currently over 97,000 works on fimfiction.net at the time of this writing. In a second pass, It would be possible to work on optimizing for speed and performance, so as to better be able to work with much larger training sets. It would also be able to examine different classifiers, such as multinomial Naive Bayes, to see what is the most effective overall.

In doing research for this project my interest was piqued in hierarchical tagging systems, and would like to see the feasibility of organizing a fanfiction website with metadata to this effect. It is clear that some genres are more related than others, like Dark and Horror. Classifying genres of fiction writing can be a fuzzy science. Better modeling how these genre tags relate to another can bring the practice into sharper focus.

References

[1] Christopher Manning. *Introduction to information retrieval*. Cambridge University Press, Cambridge, UK New York, 2008.

[2] Andrew McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI'99 workshop on text learning*, pages 1–7, 1999.

[3] Bohdan Pavlyshenko. Classification analysis of authorship fiction texts in the space of semantic fields. *CoRR*, abs/1210.5965, 2012.

[4] Wikipedia. Receiver operating characteristic — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Receiver%20operating%20characteristic&oldid=715266020`, 2016. [Online; accessed 09-May-2016].