Ruprecht-Karls-Universität Heidelberg
Institut für Computerlinguistik

Predicting Fanfiction Popularity on Archive Of Our Own

Project Report

[GitLab]

Proseminar
NLP for Social Media

Semester
SS 2018

Lecturer
Prof. Dr. Katja Markert

Author
Lyubomira Dimitrova

Email
dimitrova@cl.uni-heidelberg.de

Matr. number
3443834

October 16, 2018
Heidelberg

Table of Contents

1 Introduction

In this project I explore the problem of predicting the attractiveness of user-generated content in online communities. The content I focus on are fanfiction stories, and how much attention they receive from the online communities they are posted to.

Fandoms are the fan communities that can form around a sports team, music act, video game, or work of fiction. Fans in online fandoms are often not only consumers, but also creators of content – fanartists and fanfiction authors. An antipode to original works, *fanfiction stories* (fanfics) make use of already existing characters, settings, or even plots, to tell a story.

As of 2018, there are two main platforms for posting fan-written content – Fanfiction.net (ff.net)[1] and ArchiveOfOurOwn (AO3)[2]. A Google trends graph[3] shows that in recent years the focus has shifted to AO3. This has been attributed to the abundance of metadata in the form of tags, rating and category, which makes fanfics searchable and easily retrievable (Dalton, 2012).

Using data from AO3, I concentrate on the hit count as a measure for the amount of attention a fanfic receives. A hit is registered every time a visitor navigates to a fanfic's main page (clicks on the title)[4]. The fanfic metadata available for the prediction includes primarily the title, summary, tags and length of the fanfic. Other AO3 popularity measures, e.g. Kudos, Bookmarks and Comments, can only be registered after a hit and depend on the quality of the work itself, which is not my focus here.

The next section details some of the existing work on fanfiction analysis and popularity prediction on social media. Section 3 describes the scraping process and the final dataset, and Sections 4 and 5 define the parameters of the machine learning experiment and the model features, respectively. Sections 6 and 7 present the results and discuss findings and issues.

2 Related Work

Research on the topic of fanfiction tends to focus on literacy and writing, often in connection to L2 English learning. Some digital analyses exist (Girouard and Rubin, 2014; Milli and Bamman, 2016), though they use data from ff.net, not AO3. The closest to the aim of this project is another project report. Zhao (2016) uses deep learning to predict the review (comment) count of fanfics on ff.net,

[1] `https://www.fanfiction.net/`

[2] `https://archiveofourown.org/`

[3] `https://trends.google.com/trends/explore?date=all&geo=US&q=ao3,fanfiction.net`

[4] Some exceptions exist, like two visits in a row from one IP address counting as one hit.

using only title and summary features. Review counts are split into three bucket ranges. The LSTM achieves 72.9% accuracy on the dataset.

Predicting user attention and popularity on social media is a widely explored topic. (Hessel et al., 2017) in particular evaluate the influence of multimodal content vs context on popularity on Reddit. Since predicting the raw Reddit score of a post is challenging, they build time-controlled pairs to give compared posts an '*equal footing*'. I apply this approach to my own data, controlling not only for time, but also for topic. Section 4 details the parameters of the experiment.

3 Data

I scraped the fanwork metadata from AO3 using an unofficial Python-based scraper[5]. Thanks to AO3's advanced search some filtering of the data was possible before the scraping, along the following dimensions:

- **Only single-chapter works.** Hits are counted for each chapter of the work separately, resulting in a higher hit count for multichapter fanfics.
- **Only completed works.** Only multichapter fanfics can be ongoing.
- **No crossovers.** A crossover is a fanfic with characters from more than one fandom. I exclude crossovers from my analysis because it's difficult to tell which fanbase the fanfic will be read by.
- **Only works in English.** The vast majority of fanfics are in English, and other languages would make the feature extraction problematic.

All works were scraped 10 days after their posting date. This limit is arbitrary, but well-chosen for the larger, established fandoms. Newly emerging fandoms are generally small, and new fanfics can easily become the most popular. In older, larger fandoms, e.g. the Harry Potter fandom, newly posted fanfics are slow to gain true popularity, if they ever do.

The number of fanfics and other statistics of the final dataset were extracted in Jupyter Notebook[6] and can be found in Tables 1 and 2.

# Fanfics	# Unique fandoms	Date of posting	Average hit count
82630	5695	30.07-24.09.2018	392.7

Table 1: Dataset statistics.

[5] `https://github.com/radiolarian/AO3Scraper`. Scraping is allowed under the AO3 Terms of Service.

[6] `https://gitlab.com/lbdimitrova/ffpopularity/tree/master/notebooks/stats.ipynb`

Fandom	# Fanfics	# Pairs
Marvel Cinematic Universe	3620	1996
Voltron: Legendary Defender	3540	4165
My Hero Academia	3202	1167

Table 2: Largest fandoms.

4 Experimental Design

What is a good way to model popularity of user-generated content? Predicting a raw score like the hit count associated with a fanfic is a ranking problem – every test instance is given a score according to a ranking function learned from the training data. However, raw scores and in particular hit counts depend on many variables – when the fanfic was posted and to which fandom, which characters are used etc. For instance, a fanfic posted to a popular fandom will most certainly have a higher hit count than another posted to a less popular fandom, since the reader base is much larger. That is, only comparisons between similar fanfics are meaningful.

To solve this problem, I adopt the approach of (Hessel et al., 2017) and build time- and topic-controlled pairs of fanfics, transforming the initial data and reducing the complex ranking problem to a binary classification problem. Contrary to the approach of these authors, however, I do not discard pairs with small score differences, i.e. score differences even of 1 hit are allowed. The new data instances take the form

$$\{vector,\ label\} \ \rightarrow \ \{(f(x_i) - f(x_j)),\ sgn(hits(x_i) - hits(x_j))\},$$

where $(x_i,\ x_j)$ is a time- and topic-controlled pair of fanfics. The newly calculated labels (+1/-1) indicate which fanfic in a pair gained more attention in the form of hits. My implementation for this transformation outputs balanced classes, so that a simple random baseline has 50% accuracy.

The following time- and topic- constraints are imposed on the pairs, in order to keep the reader base as similar as possible inside a pair:

- **Both fanfics were tagged with the same relationship.** Relationship filtering is perhaps the most widely-used feature in AO3's advanced search. Figure 1a presents the average hit counts for the most popular pairings in the MCU fandom.

- **Both fanfics have the same rating**, e.g. General Audiences. Figure 1b shows the average hit counts according to rating over the whole dataset.

- **Both fanfics were posted on the same day.**

As already mentioned, this includes only single-chapter fanfics. Controlling for length is also necessary, as evidenced by Figure 1c. However, the constraint would have greatly limited the amount of data available. Instead, I use the length feature as a baseline in the experiments, in addition to the random baseline.

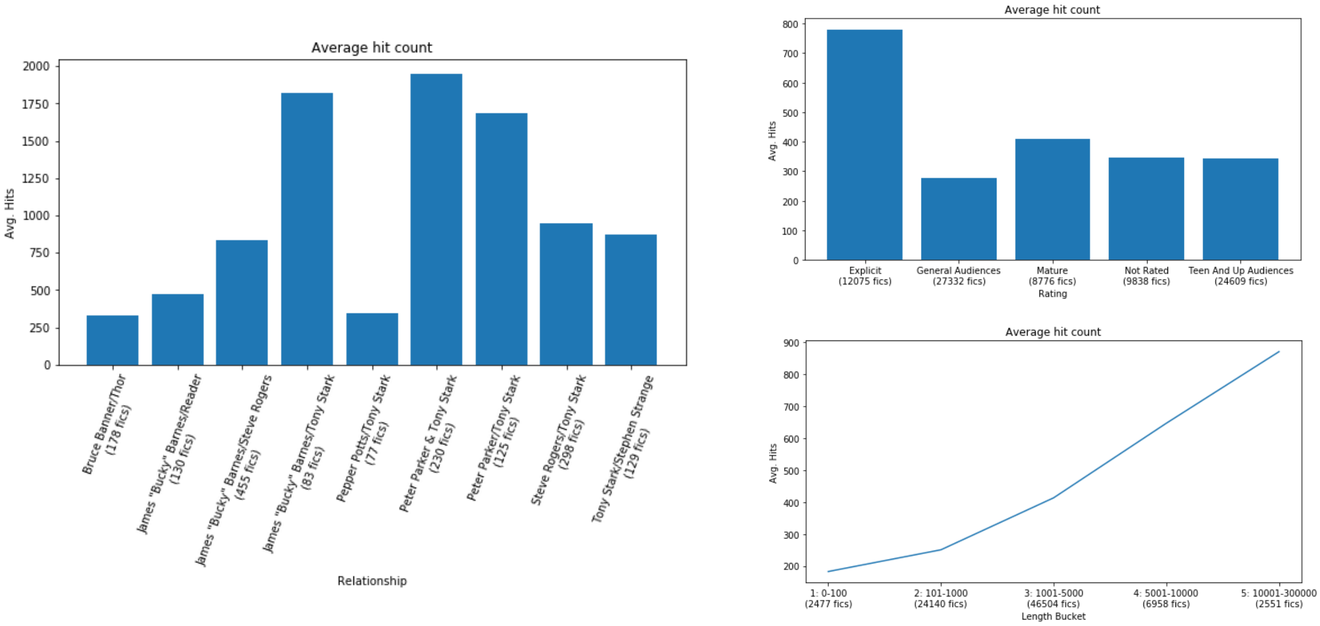

Figure 1: Average hit count according to relationship, rating, and length bucket.

5 Features

After building the topic- and time-controlled pairs, a number of differences remain between the fanfics in the pair. These include above all the summary, the title, the tags, and the length of the fanfic. The scraped metadata is in *csv* format, which facilitates the use of `pandas.DataFrame` objects in the feature extraction[7].

I use the Python library *spaCy*[8] for tokenizing and POS-tagging all summaries, titles, and tags. For the summary, I focus on structural features more than content features – length in tokens and sentences, adjective and verb ratios, entity ratios etc. Two types of readability scores are calculated, as well as stopword and punctuation ratios. Finally, I use the *TextBlob*[9] module to obtain polarity and subjectivity scores. A subset of those features are also used for the title, in addition to titlecase/lowercase/uppercase tokens ratio, and length in characters.

[7] https://gitlab.com/lbdimitrova/ffpopularity/blob/master/scripts/feature_extraction.py

[8] https://spacy.io/

[9] https://textblob.readthedocs.io/en/dev/

The use of tags is what differentiates fanfics on AO3 from those on fanfiction.net. I use two types of tag features – structural and bag-of-words, where each tag is a token in itself. The structural features include minimum, maximum and average tag length in tokens, as well as the number of tags, and the canon tags ratio. Canon tags can be used as search terms on AO3, and use Title Case – they are the official tags, which make fanworks more accessible. I cut off the BOW vectors at 300 dimensions, which limits the 'vocabulary' to the most frequent tags, usually canon ones.

Some features that would have also been interesting to explore, but which I didn't get the chance to implement, include:

- **Author Features.** Scraping AO3 proved to be very time-consuming, which had some unforeseen consequences – I had to decide against the additional scraping of author metadata. It would have been interesting to see to what extent the amount of works a fanfiction author has posted contribute to the problem of relative popularity prediction on AO3. Hessel et al. (2017) employ user features and discover that in some communities, status is a better popularity predictor than content. Unfortunately, other author statistics like the number of Followers/Subscriptions of AO3 users are only visible to the author herself, i.e. are not made public.

- **Catchy/Well-known Title.** As a user of AO3, I have noticed that many fanfics have song lyrics/poems/catchphrases as titles. Google searching title-trigrams and recording first page results could be a way to capture this tendency and explore whether famous titles cause an increase in fanfic popularity.

- **Real Events.** Although not that relevant for a single-fandom experiment, modeling real-life events could be a good predictor for user engagement in fandom. Did this franchise just release a new movie? Is that TV-show back from hiatus? Did the third book from this series just come out? Fan response to such events would be an interesting point to consider.

6 Results and Discussion

The *scikit-learn*[10] Python library provides a multitude of machine learning tools. I use a simple linear SVM model and run 5-fold cross-validation on the three largest fandoms - *Marvel Cinematic Universe* (MCU), *Voltron: Legendary Defender* (VLD) and *My Hero Academia* (MHA). The number of time-controlled pairs

[10] http://scikit-learn.org/stable/

for each fandom can be found in Table 2. Surprisingly, the VLD fandom has the largest number of pairs, most likely due to the fewer pairings in it compared to the MCU fandom (787 vs 1260).

Table 3 displays the cross-validation accuracy for different feature groups for each of the three fandoms. The performance of even more feature groups can be found on GitLab (e.g. here).

System	**MCU** (1996 pairs)	**VLD** (4165 pairs)	**MHA** (1167 pairs)
Random	50.0	50.0	50.0
Length feature	62.2	64.0	62.2
Only summary features	56.4	59.8	60.5
Only title features	53.0	53.1	47.4
Only tags features (no BOW)	55.2	57.2	57.5
Only tags BOW features	65.4	68.0	65.5
All tags features	66.2	69.0	64.0
All features	**68.1**	**70.1**	**66.6**
All features (score difference >= 20)[11]	71.3	71.3	68.3

Table 3: CV Accuracy. Bolded scores are the highest.

As expected, the length baseline performs very well, outperforming the summary, title, and simple tags features for all three fandoms. Only the last feature group - the bag-of-words tags features - manages to score an improvement over this baseline. The 'All features' system yields the best results, reaching a 6% improvement over the length baseline for the VLD fandom. Apart from the feature groups in the table, this model also includes category, character and relationship features, which generally performed about as well as the title features.

The results also show that tags have the best predictive power when it comes to hit count, and adding in all other feature groups only results in a $\sim$ 2% improvement.

Some fandom differences become apparent - the summary and simple tags features perform better on the VLD and MHA fandoms, while the title features are weaker, and even perform worse than the random baseline on the MHA fandom.

The last row displays the improvement when difficult to classify pairs are discarded. The hit count difference cut-off is set at 20 because higher values cause a large data loss.

All in all, the length feature is a very good predictor for relative popularity of fanfics on AO3. The bag-of-words features also perform extremely well considering their simplicity, which confirm the findings of (Hessel et al., 2017) that "*For Words, Simpler is Better.*"

[11] The score difference is calculated when the pairs are build. This cut-off produces fewer pairs.

7 Conclusion

I scraped data from the most popular platform for fan-written content and designed a machine learning experiment to evaluate whether predicting fanfiction popularity only from metadata is possible.

The results indicate that a larger amount of data would be beneficial for improving classifier accuracy. Other improvements I would suggest are including more features, like the ones mentioned in Section 5, and using deep learning following the approach of (Zhao, 2016).

This project was something entirely new for me and definitely challenging. One of the issues I faced is that AO3 has no official API. The unofficial scrapers available on GitHub had some functionality in excess and were missing some necessary features. Tweaking the scripts and debugging the new code was time-consuming. Another problem was that I severely underestimated the time needed to scrape a significant amount of data from AO3. The daily scraping of $\sim$ 1600 fanfics lasted around 3.5 hours. My initial idea was to conduct a multi-fandom experiment and build the pairs with an additional fandom constraint. However, I had some hardware difficulties that made the feature extraction for all 82630 instances impossible, which led me to limiting the experiment to one fandom at a time.

I learned a lot working on this project, perhaps because of these difficulties. A lot of work goes into designing features and making sure every code block works exactly as intended. Many ideas were thoroughly researched, but not implemented. All things considered, I found this experience good preparation for future software projects.

References

Dalton, K. L.
2012. Searching the archive of our own: the usefulness of the tagging structure.

Girouard, V. and V. L. Rubin
2014. Comparative stylistic fanfiction analysis: Popular and unpopular fics across eleven fandoms. In *Proceedings of the Annual Conference of CAIS/Actes du congrès annuel de l'ACSI*.

Hessel, J., L. Lee, and D. Mimno
2017. Cats and captions vs. creators and the clock: Comparing multimodal content to context in predicting relative popularity. In *Proceedings of the 26th International Conference on World Wide Web*, Pp. 927–936. International World Wide Web Conferences Steering Committee.

Milli, S. and D. Bamman
2016. Beyond canonical texts: A computational analysis of fanfiction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Pp. 2048–2053.

Zhao, A.
2016. Predicting popularity of fanfiction stories based on title and summary.