Link Prediction in FanFiction Networks: Early Work Toward Collaborative Filtering

Alexander L. Hayes
The University of Texas at Dallas
alexander.hayes@utdallas.edu

Abstract

The author performs link prediction on metadata and social network data scraped from FanFiction.Net (the world's largest repository of user-submitted fanfiction), applying statistical relational learning to predict authorship of a story through two tree-based models: relational dependency networks (RDNs) learned via gradient boosting, and RDNs learned via gradient boosting with soft-margin constraints set on the cost of false positives and false negatives. Results are shown for four types of fanfiction, and transfer is applied to show how structure learning generalizes across different social networks. Code and documentation for the experiments are available in the Appendix section of this paper and on GitHub.[1]

Introduction

A "fanfic" is a creative work where a person adapts or extends an existing work of fiction. An author may write a completely different ending to the source material, he or she may tell a similar story but insert several characters of their own creation, or he or she may cut the characters from one story and place them in a completely different environment. What if the characters from Tolkien's *The Lord of the Rings* were in a modern-day high school? What if Jean-Luc Picard landed the USS Enterprise on *Avatar*'s Pandora? What if *Phantom of the Opera* had a completely different ending?

Scenarios such as these are played out on places such as FanFiction.Net, where authors and community members write, read, and review one another's stories across thousands of possible fandoms. The practice of writing and sharing fanfiction is sometimes compared to oral-traditional storytelling; rather than media being produced and distributed by a few popular people or publishing groups: stories are shared, retold, and adapted between a community of people; building universes and mythos around stories.

FanFiction has been observed through a variety of academic lenses; such as psychology, sociology, and (more recently) computer science. Though many have taken interest in the subject, a common viewpoint is that fanfiction is poorly understood within academic literature, or that previous attempts to analyze fanfiction communities have been performed in disruptive manners (Larsen and Zubernis 2011). (Barnes 2015) noted that the subject at the time of writing had almost entirely been understood from a qualitative standpoint, but posed several questions which may be answered in part by large-scale quantitative analysis.

[1] https://github.com/batflyer/FanFiction-Collaborative-Filtering

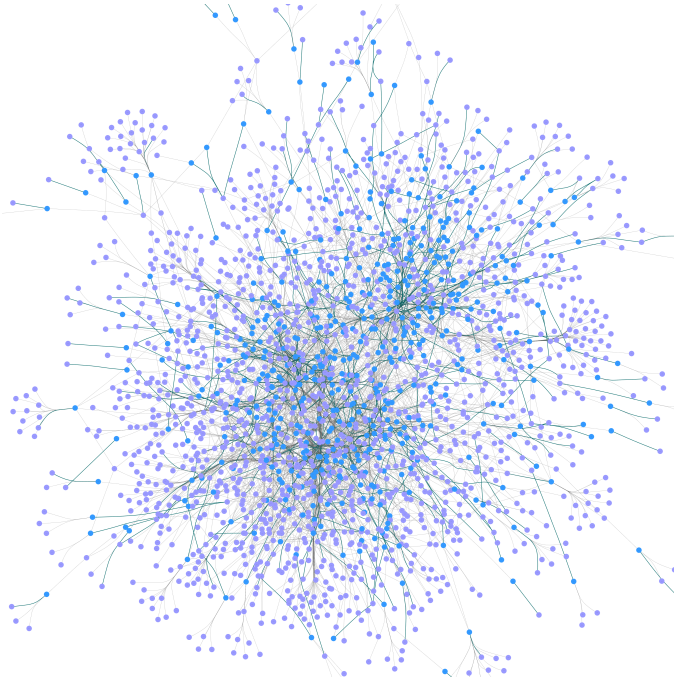

Figure 1: Network of Coraline fanfiction. Blue nodes represent stories, violet nodes represent users. Darker edges represent authorship, lighter edges represent a review.

Related Work

Within the last few years has there been an increase in large-scale computational analyses. (Milli and Bamman 2016) compared fanfiction with the source (canon) text that stories originated from, performed sentiment analysis on the reviews, and predicted how readers would respond to characters in a chapter.

(Yin et al. 2017) published an anonymized set of meta-data, and presented evidence that the majority of users were English-speaking students based on the time of year FanFiction.Net users were most active. This partially answered one of the questions posed by (Barnes 2015), "who writes fanfiction?" While this meta-dataset may give valuable insight to many problems, it also eliminates one of the most valuable properties: the social network structure which exists between users and stories.

Collaborative Filtering

Document search is often performed as a combination of two components: *content filtering* to find relevant keywords, document attributes, or metadata; and *collaborative filtering* for evaluating relevancy through a popularity or similarity metric.

Collaborative filtering in networks or graph models may be performed in supervised or unsupervised manners for determining local, quasi-local, or global similarity (Nickel et al. 2016; Breese, Heckerman, and Kadie 1998).

Unsupervised global similarity methods such as PageRank may provide insight to what is popular across all users in the a network, but may be less insightful when certain parts of the network are only relevant to certain users. Local similarity methods such as random walks may focus on parts of the network which are most relevant to certain users, but may not reach far enough outside of what the user is already aware of.

Supervised methods for collaborative filtering turn the previous into a learning problem where the similarity between two entities or relationships may be inferred from some observed properties of the data. These methods come with all of the difficulties of machine learning, such as balancing overfitting against generalization; but a prominent difficulty of these methods lies in the imbalance between the false positives and false negatives. A recommendation system that recommends everything to a user rather than tailoring results to their interests is not a good recommendation system.

(Yang et al. 2017) recently applied statistical relational learning methods to a large-scale job recommendation system, using both content filtering and collaborative filtering on four months of data to recommend jobs to a user in a supervised manner. They found that changing the cost function to put a higher weight on false positives led to better results. Inspired by this work, we build on the recommendation system aspects but apply it to predicting authorship of a story and evaluate whether the structure learned in one network may easily be transferred to another.

Experiments

The author poses and answers the following questions:

Q1: How effective are the models at generalizing within and across communities when predicting story authorship?

Q2: Does a greater number of facts in the training set directly lead to better results on the test set?

Q3: Does soft margin improve performance over the default settings?

Creating a FanFiction Data Set

All authors and stories on FanFiction.Net may be identified by a unique integer, which naturally lends itself to a relational representation. The meta-data about each story includes attributes such as the number of words, number of chapters, main characters, genres, and rating. Stories may be introduced with a summary in fewer than 384 characters, but these are not included here. Stories may be divided into chapters, and chapters may be reviewed by the author or other community members.

Four communities on the FanFiction.Net books category were selected for these experiments: *Coraline*[2], *Hitchhiker's Guide to the Galaxy*[3], *To Kill a Mockingbird*[4], and *Dragonriders of Pern series*[5].

Data was scraped from these communities between April 24 and April 27, 2018, and four features were chosen to learn and make predictions about story authorship: `author` (the target attribute), `genre`, `rating`, and `reviewed`. `author` is a binary predicate distinguishing that a user wrote a story. `rating` is an attribute describing the fiction rating of a story (Li 2005). `genre` describes the genre of a story (Adventure, Horror, etc.), a story may have up to three genres but for simplicity all genres for a story are stored as a single value. `reviewed` is a binary predicate indicating that a particular user reviewed a particular story. These features were chosen under the assumption that in these communities:

- Authors are more likely to write stories with similar genres and ratings: an author tends not to write for all genres with equal frequency.
- Users tend to review stories with some genres or ratings as opposed to others, and personal preferences dictate what those may be.
- Users are more likely to review stories by the same author, either due to popularity of a story or an author's network of followers.

Though more features (such as word count, correlation metrics, review sentiment) may be vital for deeper insights, these four are appropriate to establish baselines and gain some insight to the prior assumptions.

These communities were selected because they shared a similar number of stories (between 575 and 625). Though the communities were similar by this metric, there was a prominent difference in the number of facts: *To Kill a Mockingbird* had the highest number of stories, but the fewest facts; *Dragonriders of Pern series* had the second-fewest number of stories, but significantly more facts. This is because the number of facts associated with a community is heavily biased by the number of reviews across a community's stories. This may be a way to gain insight into community activity in subsequent studies.

[2] https://www.fanfiction.net/book/Coraline/

[3] https://www.fanfiction.net/book/Hitchhiker-s-Guide-to-the-Galaxy/

[4] https://www.fanfiction.net/book/To-Kill-a-Mockingbird/

[5] https://www.fanfiction.net/book/Dragonriders-of-Pern-series/

		Learning		Inference	
Community	# facts	pos	neg	pos	neg
Coraline	4,981	402	160,873	173	29,215
Dragonriders	6,678	421	174,607	179	31,833
Hitchhikers	5,240	423	177,362	180	32,322
Mockingbird	3,793	438	190,051	187	34,563

Table 1: The number of positives, negatives, and facts for each community during learning and inference. For evaluation, `author` values were randomly assigned to a learning or inference set based on a 70%/30% train/test split, and these splits were sampled and averaged ten times.

Learning and Inference with BoostSRL

Facts were made available during both learning and inference whereas positive examples were assigned randomly to a learning or inference set to make predictions. Negative examples were sampled under the closed-world assumption from the respective positive set, but because the number of negatives is determined from the positives: the number reported for each learning and inference set is an average from 100 samples.

The author employs a state-of-the-art statistical relational learning system: *BoostSRL* to learn relational dependency networks over the features. Two methods are compared: boosted relational dependency networks (RDN-Boost), and boosted relational dependency networks with soft margin (Softm-Boost). For both methods, the learning and inference sets were sampled from master sets for each community, and precision/recall scores were calculated based on an average over ten independently sampled runs. To evaluate how general these models are, the learning and inference sets may be drawn from two different communities: learning may be done on a set of stories drawn from *Coraline*, but inference may be performed on stories from the *Hitchhiker's Guide to the Galaxy* community.

For both RDN-Boost and Softm-Boost, all settings were kept constant, including: tree depth, node size, mode settings, and number of regression trees learned during training (15). The only exception was the use of soft margin with false negative (α) weight of 0 and a false positive (β) weight of 2 in order to increase the cost of returning false positives.

(Yang et al. 2017) used similar settings to these and claimed that learning more than 20 trees had little affect on overall performance. Varying these settings with careful tuning may lead to better performance, but this possibility is not explored within the context of this paper.

Results

Table 2 displays the results of the experiments. Soft margin leads to a massive improvement in recall across all training and test splits, but precision appears to be negatively affected in some cases. **Q3** may be accepted in this case: if the goal is high recall, then the soft margin approach should at least be explored.

If the number of facts available during learning lead to better results on the given test set, it would generally be assumed that the set with the most facts would lead to the best performance while the set with the fewest facts would lead to generally less-good performance. In this case, *Dragonriders of Pern series* had the highest number of facts (6,678) while *To Kill a Mockingbird* had the fewest (3,793). For both RDN-Boost and Sofm-Boost, the best performance tends not to be based on the number of facts available during learning or inference, but whether learning and inference draws examples from the same community. This provides insight to answering both **Q1** and **Q2**: more data does not necessarily imply better results when the underlying data is drawn from different distributions; but the performance does not drop in such a catastrophic manner to make us assume that generalization does not occur at all.

Conclusion

These results are still fairly early in the process toward something which could become a collaborative filtering engine for FanFiction.Net, or a full paper extending what is known about several communities on FanFiction.Net. The evaluation criteria presented here were measured against what is an extremely difficult task of predicting authorship from a small number of features. In addition to the open problems left over from these results, some of the key takeaway messages should be these: (1) FanFiction.Net has rich social network properties which should be further explored within the statistical relational learning community, and the computer science community more broadly. (2) FanFiction.Net is the embodiment of a time-evolving social network, where people who share common interests read, write, and review stories.

Appendix

Scraping FanFiction.Net

The terms of service for FanFiction.Net state the limitations of launching automated bots in Section 4, Paragraph E of their terms of service[6], paraphrased here:

- "You agree not to use or launch any automated system... that accesses the Website in a manner that sends more request messages... than a human can reasonably produce in the same period...."
- "FanFiction.Net grants the operators of public search engines permission to use spiders... solely to the extent necessary for creating publicly available searchable indices... but not caches or archives...."
- "You agree not to collect or harvest any personally identifiable information... for any commercial solicitation purposes."

Scraping FanFiction.Net may be accomplished via the `ffscraper/` Python package in this repository. The software is distributed under an Apache-2.0 License and the core documentation pertaining to each script and their respective functions are contained therein. This discussion pertains to the high-level idea of what they accomplish.

All users and stories on FanFiction.Net may be identified via a unique id-number. These id-numbers may be

[6] https://www.fanfiction.net/tos/

RDN-Boost Precision				
	Coraline	Dragonriders	Hitchhikers	Mockingbird
Coraline	**0.014**	**0.009**	0.013	0.012
Dragonriders	0.008	**0.009**	0.009	0.013
Hitchhikers	0.013	0.008	**0.019**	0.014
Mockingbird	**0.014**	0.007	0.013	**0.034**
RDN-Boost Recall				
	Coraline	Dragonriders	Hitchhikers	Mockingbird
Coraline	0.119	0.092	0.062	0.065
Dragonriders	0.037	0.083	0.063	0.050
Hitchhikers	0.206	0.227	0.193	0.075
Mockingbird	0.331	0.415	0.258	0.210
Softm-Boost Precision				
	Coraline	Dragonriders	Hitchhikers	Mockingbird
Coraline	**0.014**	**0.011**	0.013	0.001
Dragonriders	0.009	0.010	0.008	0.008
Hitchhikers	0.010	0.008	**0.023**	0.016
Mockingbird	0.009	0.008	0.015	**0.025**
Softm-Boost Recall				
	Coraline	Dragonriders	Hitchhikers	Mockingbird
Coraline	1.0	1.0	1.0	1.0
Dragonriders	1.0	1.0	1.0	1.0
Hitchhikers	1.0	1.0	1.0	1.0
Mockingbird	1.0	1.0	1.0	1.0

Table 2: Precision and recall values for RDN-Boost and Softm-Boost. Rows represent the learning set, columns represent the inference set. Precision values are highlighted if they are the highest in either a row or column.

used to directly access one or the other, for example: user *124*'s profile may be found by sending a request to https://www.fanfiction.net/u/124. In a similar manner, story *124* would be at https://www.fanfiction.net/s/124. Reviews for a story follow the pattern `/r/125/0/1/`, where the first integer corresponds the story-id, the second integer to a certain chapter (or 0 for all reviews), and the third is a page number (a maximum of 15 reviews are displayed on each page, so any overflow will be on subsequent pages).

References

Barnes, J. L. 2015. Fanfiction as imaginary play: What fan-written stories can tell us about the cognitive science of fiction. *Poetics* 48:69–82.

Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 43–52. Morgan Kaufmann Publishers Inc.

Larsen, K., and Zubernis, L. 2011. *Fandom at the crossroads: Celebration, shame and fan/producer relationships*. Cambridge Scholars Publishing.

Li, X. 2005. Fiction Ratings. https://fictionratings.com.

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

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1):11–33.

Yang, S.; Korayem, M.; AlJadda, K.; Grainger, T.; and Natarajan, S. 2017. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive statistical relational learning approach. *Knowledge-Based Systems* 136:37–45.

Yin, K.; Aragon, C.; Evans, S.; and Davis, K. 2017. Where no one has gone before: A meta-dataset of the world's largest fanfiction repository. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 6106–6110. ACM.