Evergreen or Ephemeral: Predicting Webpage Longevity Through Relevancy Features

Elaine Zhou, Lingtong Sun
Stanford University, Stanford, CA 94309
{ezhou, lsun569} @stanford.edu

I. INTRODUCTION

With the rapid proliferation of user-generated content available on the Internet, one of the biggest challenges is determining the relevancy of the information shown. The content often comes in two camps: ephemeral or evergreen. Evergreen content such as recipes for carrot cake or intro to data structures frequently don't change with time, whereas ephemeral content, such as celebrity hot or not trends or local high school sport scores easily become dated. Unlike apps that harp on ephemerality like Snapchat, the Internet doesn't have the luxury of assigning expiration dates to content. Humans can easily distinguish one from the other, but machines have yet to do so.

The challenge here is to predict whether or a not a new piece of web content will be ephemeral or evergreen. This would be helpful for all sorts of recommenders to sift through relevant content, improving web search results, or customer opinion or review pieces, or general methods for prioritizing web archives. Our goal is to develop a prediction model and identify most relevant attributes for evergreen-ness using machine learning techniques.

II. BACKGROUND & DATA SET

StumbleUpon is a user-curated web content discovery content that aims to recommend relevant media and links to its user base. At the crux of improving their recommendation engine is the problem of ephemeral or evergreen content -- how can we try and classify websites before voted on by users?

Thus, as presented by Kaggle, we use StumbleUpon's raw HTML content scraped from over ~10,000 websites, a training set of evergreen or not 7,395 labelled URLs, and a test set of 3,171 unlabelled URLs [1]. These websites were compiled in a .tsv file with aggregated text and 24 numeric meta-data fields, such as the boilerplate text, the ratio of spelling errors, or ratio of <img> tags vs text. The boilerplate was a JSON string format with the title, body text, and url of the website [2].

III. FEATURE SETS & PRE-PROCESSING

First, we set out to understand the nature of the data given and how previous StumbleUpon users classified evergreen websites. At an initial glance, the numeric data provided by Kaggle fell in two camps: those related to amount of embedded media (e.g. embed_ratio, html_ratio, image_ratio) and the content of embedded media (e.g. numwords_in_url, commonLinkRatio_1, linkwordscore). There was very little related to the interactivity from javascript code or the markup of body text itself from HTML tags. Previous research on web page longevity validated our initial observations -- although there is much variation among web page in terms of top-level domain and by page type, web content itself is more likely to stabilize [3]. The insight here is that we should focus more on the content and less on the HTML structure or DOM itself.

Next, we dove deeper into the text data provided by the page url, body text, and title. We initialized our text feature selection by using the boilerplate of the elements, but immediately found some issues. There was empty body text, empty titles, and the body and title text often were rife with advertisements. Having any sort of article with a major component as null or empty labelled as evergreen can be seen as a false positive. In our preprocessing procedure, we cleaned up this data, as we later discuss.

From the numeric and text data provided and our observations and intuitions about HTML webpages, we proceeded to focus on our features in two parts: the 25 numeric meta-data values (with the addition of converting alchemy_category to an integer

representation), and 3 text features given by the boilerplate: title, url, and body text.

iV. DISCUSSION & RESULTS

We discuss our evaluation and use of several different types of classifiers, and our research into the different feature sets aforementioned and their effectiveness.

Before we delve into the specifics of Classifier and Feature Selection, it is important to discuss the scoring method that we employed. For each feature set and classifier, we implements K-folds Cross Validation with a k value of 10, meaning we divided our training data into 10 subsets and ran the classifier 10 times (each time using a different subset as the test set and the rest as training sets). The score for the classifier and feature set was then determined to be the mean of the cross validation.

A. Classifier Selection

We now discuss the three main classification algorithms that we investigated in detail: Naive Bayes, logistic regression, and support vector machines (SVMs). We began with the numeric meta-data provided to get a better understanding of the problem at hand, but we iterated and improved on these models by adding the text features and other feature sets.

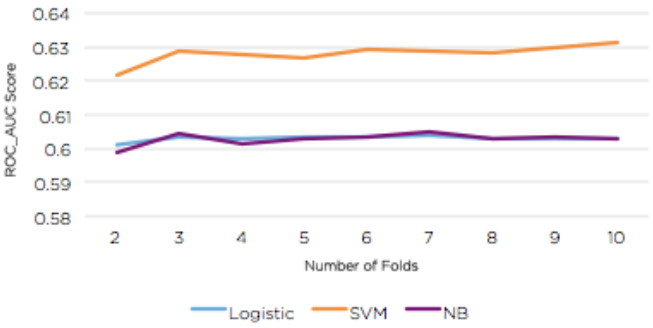
1. *Naive Bayes*: As recommended in class lecture, we began with an implementation of the Multinomial Naive Bayes Classifier. We focused strictly on the content, and just used the alchemy_score and image_ratio as a quick baseline.

2. *Logistic Regression*: From the Naive Bayes exercise, we realized that this model tended to underfit the data. To reduce the bias, our next attempt was with the logistic regression. The difference in scores was only marginal.

3. *SVM*: We implemented SVM and found that C = 0.5 gave the best result for the data model. Quick note on other models: We also attempted linear regression and quadratic discriminant analysis models, but their yield wasn't close to that of the Naïve Bayes, Logistic Regression, or SVM, so those results are not included in this report. We decided to keep iterating on these as main models for our dataset.

We show the results of these three classifiers with K-Folds Cross Validation (Figure 1). In most, we found there to be a fairly high bias (since we aren't using many features but it gave us a good understanding on which models to focus on.

Figure 1: K-Folds Cross Validation for Logistic Regression, SVM, and Naïve Bayes



B. Feature Selection

From our first implementation of a number of models, we were getting similar results without much improvement. After some more exposure to the dataset, We decided to expand on the input features and focus on the text data features as well. We began by letting machine learning algorithms choose the best features to select through forward feature search.

1. *Forward Feature Search*: Our rudimentary analysis of 6 features led up to mediocre results, so we wanted to let the data choose the features instead. We maintained our three classifiers (naive bayes, logistic regression, and SVM) and ran forward feature search for the top 5 features. The features that appeared most in all the models are: linkwordscore, numberOfLinks, and numwords_in_url. However, even with the choice of all 25 different features, our improvements, feature over feature, were not significant. From Figure 2, we can see that for logistic regression and naive bayes models, we quickly approach a limit of a score of around 0.61. The second realization we had is that features we were provided were not the most helpful -- after 2 or 3 of the best performing features, the score peters out. The maximum features and the resulting subsequent score is shown in figure 3. Thus, our next step was to gather more informative features and we turned to parsing and understanding our text features.

Figure 2: Changes in scores by adding best features in each classifier

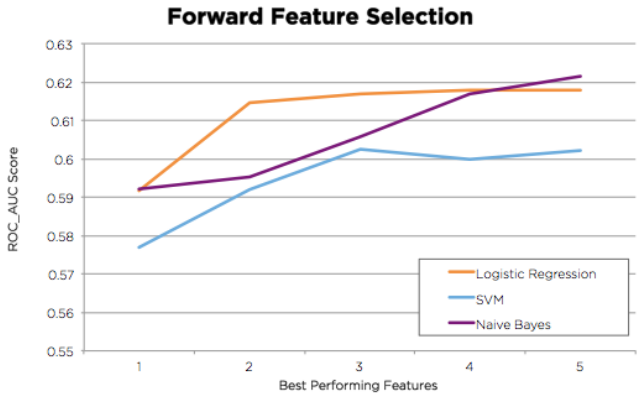
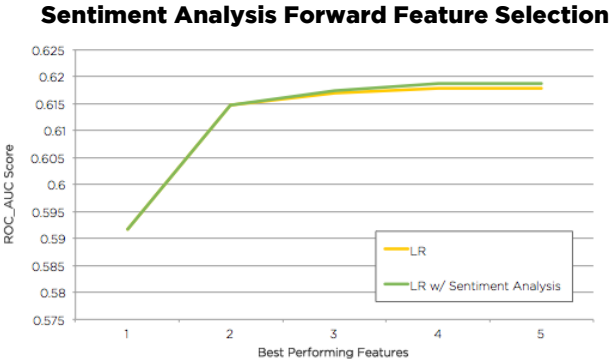


Figure 3: Best Features from Forward Feature Search for various models

Logistic Regression		SVM		Naive Bayes	
Top Feature	*Score*	*Top Feature*	*Score*	*Top Feature*	*Score*
linkswordscore	0.59171	linkwordscore	0.57684	linkwordscore	0.59216
numberOfLinks	0.61469	image_ratio	0.59214	lengthylinkdomain	0.59531
non_markup_alpha numeric_character	0.61694	alchemy_category_score	0.60252	numwords_in_url	0.60568
embed_ratio	0.61785	numwords_in_url	0.59982	numberOfLinks	0.61694
commonlinkratio_1	0.61785	is_news	0.60207	is_news	0.62145

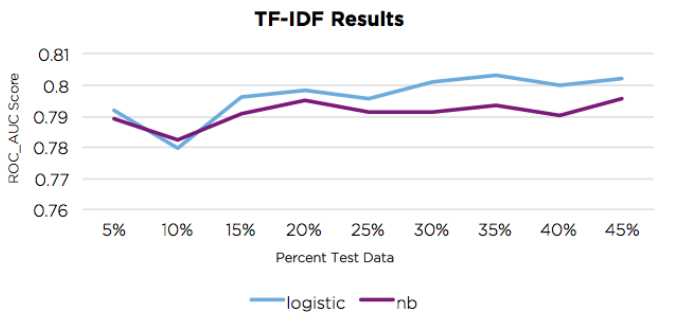
2. *Sentiment Analysis*: Our background research and forward feature search indicated that body text was critical in understanding website longevity. From our conversations on how best to represent this information, we moved to sentiment analysis. We hypothesized that highly intense, emotional reactions or highly negative or positive text might affect the ephemerality of a website. Using the open library TextBlob, we added: the title subjectivity, title polarity, boilerplate body text subjectivity, boilerplate body text polarity, and url subjectivity, and url polarity. This resulted in 6 new features to our forward feature search. Both the subjectivity of the title and the sentiment of the body paragraph appeared in the top 5 features in the logistic regression model. However, our hypothesis was not fully realized -- adding these features improved our maximum correctness rate by a marginal difference. As shown in Figure 4, the logistic regression score improved 0.1%. Upon further examination, we realized there was insufficient text data from the boilerplate the body text could be found in the boilerplate JSON. Next, we tried to expand beyond the boilerplate data to include all the body text data from the websites. Using a scraper, we repeated the process but with the full text of the websites with python-boilerpipe, a python wrapper for the Java HTML fulltext extraction library [5]. Using the top feature determined from forward feature search, we added the fields for body text subjectivity and body text polarity. However, this was unfruitful after much processing power and wait time, as our accuracy dipped. Ultimately, there was too much noise in the body text.

Figure 4: Changing Score With Sentiment Analysis



3. Term Frequency-Inverse Document Frequency (TF-IDF) Features: After trying the aforementioned sentiment analysis, forward feature search, and various other techniques to improve the performance of our classifier to limited success. We did some further research and determined that the text frequency of the site contents might yield more promising results. The feature we decided to extract is the Term Frequency - Inverse Document Frequency (tf-idf). Essentially, tf-idf is the term frequency times the inverse frequency. The advantage of using tf-idf to term frequency alone is that we can scale down the impact of tokens that occur very frequently and thus give us less classifying information. The formula uses a form of laplace smoothing in that we do tf * (idf + 1) to not entirely ignore terms with 0 idf. We employed SciKit's built in tf-idf extractor on the content of our websites and ran logistic regression as well as our naive bayes classifiers with this new feature set. We obtained a much better classification accuracy using the same scoring system described earlier. With Logistic regression, we obtained an accuracy of 0.8754; with NB, we obtained an accuracy of 0.8679.

Figure 4: TF-IDF Results



When combined with our earlier results, tf-idf seems to drown out the other features and dominate in the classification because adding the other features did not alter the results of classification at all. However, with tf-idf, we were able to build a much more successful classifier.

V. CONCLUSION

Understanding the format and content of a website's body text is vital to many web-mining applications and we presented a method to effectively solve the StumbleUpon Evergreen Challenge. However, despite given the emphasis on the body content, it is important to filter out the noise and spam from a variety of sources to present stronger classification features.

VI. FUTURE WORK

There is still much to be learned in the topic of website longevity. Future work would involve gathering more and different data sources. As we learned, the content of the website is critical is classifying its status as evergreen. However, we are definitely lacking in the types of raw data that StumbleUpon contains If we take the alchemy category as a basis, the first thing we recognize is that there are no articles under the category "religion", indicating there are article areas that are lacking. Considering the timelessness of religious texts like the Bible or the Quran. Neither author has any expertise in this field, but in an analog example, 100m copies of the Bible are sold or given away each year and the Quran is one of the most widely read and recited book in the world [4]. Expanding data sets into more unconventional sources might yield further insights into website ephemerality.

Another potential exploration could be in the labelling of evergreen or ephemeral from the get-go. It is not explained how these were determined -- whether it was the opinion of an individual or of a group. Either way, it's not the best reliable measurement of website longevity since it has an inherent bias towards the interests of StumbleUpon's user demographic. As our data perusal showed, hand-labelling is also prone to error. Perhaps StumbleUpon can implement another metric to supplement or quantify the label of "evergreen", maybe to track the change or stability in traffic over time, or the click-through rate of articles when presented to real end users. These are just a few changes that could help establish and explain evergreen while reducing the rate of incorrect labels.

VII. REFERENCES

[1] Kaggle, "StumbleUpon Evergreen Classification Challenge", http://www.kaggle.com/c/stumbleupon

[2] Kaggle, "Data - StumbleUpon Evergreen Challenge", https://www.kaggle.com/c/stumbleupon/data

[3] Koehler, W., "Web Page Change and Persistence—A Four-Year Longitudinal Study"

[4] The battle of the books http://www.economist.com/node/10311317