Classifying Ephemeral vs Evergreen Content on the Web

Li-Wei Chen (lwc@stanford.edu)

I. Introduction

ONE of the strengths of the internet is the proliferation of content available on virtually any topic imaginable. The challenge today has become sorting through this wealth of content to locate the information of greatest interest to each user. Many sites today implement recommender engines based on expressed and learned user preferences to direct users towards new content that the engine believes they will most enjoy. The relevance of such content can either be highly topical and short-lived (such as last night's sports scores) or enduring and long-lived (such as an introductory tutorial to machine learning algorithms). The former content is termed "ephemeral", while the latter is called "evergreen".

An interesting challenge is to attempt to predict a priori if a new piece of content will be in the former or latter category. Not only would it be useful for recommenders attempting to classify different news stories based on type, this information could be used for other applications also, such as for archival projects to determine what web content merits inclusion, or for content sites interested in capacity planning for hosting different pages based on expected longevity.

II. Project Design

A. Dataset

This project uses the dataset provided by StumbleUpon as part of the "StumbleUpon Evergreen Classification Challenge" competition on Kaggle [1]. The dataset consists of a training set of 7,395 URLs that have been hand-labelled as evergreen or not, and an unlabelled test set 3,171 URLs. We evaluate the performance of several different classification algorithms in accurately predicting the evergreen status of different pages.

The metric for evaluation used will be the one chosen by Kaggle for the contest: the area under the receiver operating characteristic curve (ROC AUC) [2] [3]. The ROC is a characterization of the true positive rate against the false positive rate of a classifier. The area under the ROC curve is equal to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

B. Feature Selection

The basic information available for each training example is the URL of the page. Along with the URL itself, Kaggle provides a snapshot of the HTML retrieved from the URL in raw form. The first challenge is to transform the HTML page into features that can then be processed by classification algorithms.

HTML pages today include much more than the content text of the main topic of the page. As an illustrative example, consider the page http://www.howsweeteats.com/2010/03/cookies-and-cream-brownies/, which is an example of an evergreen page for a brownie recipe drawn from the training data. The page consists of the recipe itself, some anecdotal descriptions about the author's experiences baking with the recipe, and photographs of the food in question, all relevant to the interest level of the viewer to the page. However, it also contained a drug ad and an automotive ad, as well as verbose javascript implementing a user tracking system, which is likely not relevant to user interest, as well as generic items such as a commenting system and links to various locations on the parent site which, while they might contain relevant content, are common components of both evergreen and non-evergreen sites.

Some basic intuition on preprocessing the HTML to perform feature extraction can be obtained by training a regularized logistic regression classifier on the raw HTML and examining the words with the lowest predictive weight. One insight from this exercise is that the raw tags themselves contain very little predictive power, likely because they appear in virtually all the documents. Similarly, the javascript code was also typically not related to the document contents and not predictive. Preprocessing the HTML to strip the tags and javascript and keep only the contents of the tags themselves both reduced the amount of data that the algorithms needed to process as well as reducing the noise in the input.

Some of the features that were included in this project based on their predictive possibilities:

- "url": Page URL
- "body": Body text
- "links": Body href links
- "outline": Title and header node contents

C. Preprocessing

The extracted features were preprocessed to transform them into feature vectors. First, the contents of each page were transformed to standard ASCII encoding. The body text and title and header node contents were common English-language words, and were stemmed using a Porter stemmer [4]. The URL features were "stemmed" by extracting the domain from each URL. The bag-of-words model was applied to the stemmed text and domains [5].

Two approaches were used to transform the resulting bag-of-words data into input features for the classification algorithms. The first computes a document-term matrix where the rows correspond to different training examples, and the columns

indicate the term frequencies of the different words in the dictionary. We construct the dictionary by computing the term frequencies of all words appearing in the entire training set, and discarding the most and least frequently appearing words. The rationale for this approach is to discard the filler words in the English language (such as "a" or "the") which have high frequency but little information, and also the very low-frequency terms which do not occur often enough to be generally useful for prediction.

In the second approach, we use the term frequency-inverse document frequency (tf-idf) of each word. The tf-idf is the product of the term frequency, indicating the number of times a word appears in a given document, and the *inverse document frequency*, which measures how commonly the word appears across all documents. The inverse document frequency is computed as

$$\mathbf{idf}(t, D) = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \quad (1)$$

where D is the set of training examples (documents), $|D|$ is the number of training examples, and $|\{d \in D : t \in d\}|$ is the number of documents where the word t appears [6]. The inverse document frequency will be small when the same term appears in a large proportion of the documents, and multiplying it into the term frequency will decrease the weighting on terms that appear commonly in the majority of documents (and thus are unlikely to have much predictive power).

III. Results

In this section, the performance of several different types of classifiers is evaluated. The predictive potential of each of the feature sets is also investigated.

A. Classifier Selection

Three different classification algorithms were investigated: Naive Bayes, regularized logistic regression, and support vector machines (SVMs).

The Naive Bayes classifier was trained on the document-term frequency matrix. The dictionary was sorted in order of term frequency, and varying numbers of the most and least frequent words were discarded to investigate the effects of trimming the dictionary. Trimming removed words with low predictive strength from the dictionary and helped prevent overfitting on noisy data.

Table I shows the ROC AUC results evaluated via cross-validation. It indicates that while trimming low-frequency words degrades the metric, trimming high-frequency words has little to no impact. This suggests that some of the infrequent words do have predictive power so there is value in retaining them, but that the high-frequency filler words can be discarded without penalty. This would be useful for controlling the size of the dictionary and reducing run times for a classifier based on Naive Bayes that was being used in a production environment.

Figure 1 sheds more light on the effect of trimming the dictionary. The divergence between the training and test error for the full dictionary shows that Naive Bayes is overfitting due to the addition of the low-content words. The training and test learning curve convergence indicates that the removal has reduced the overfitting (although it has not resulted in an improved score).

TABLE I. Naive Bayes cross-validation ROC AUC on trimmed data

% trim	90	95	98	99	100
0	0.830	0.830	0.830	0.830	0.829
1	0.821	0.821	0.821	0.821	0.821
2	0.816	0.816	0.816	0.816	0.816
5	0.811	0.811	0.811	0.811	0.811
10	0.806	0.807	0.808	0.808	0.808

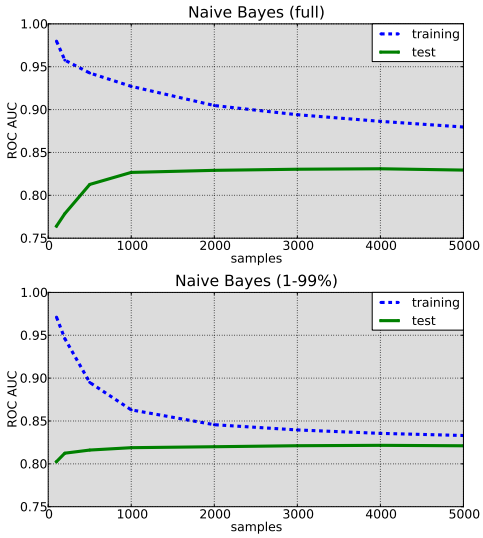

Fig. 1. Learning Curves for Naive Bayes

TABLE II. Logistic regression cross-validation ROC AUC on trimmed data

% trim	80	90	95	100
0	0.793	0.794	0.793	0.793
5	0.776	0.778	0.778	0.778
10	0.810	0.811	0.811	0.811
15	0.815	0.816	0.817	0.817
20	0.815	0.816	0.817	0.817
25	0.810	0.812	0.812	0.812

Table II shows the ROC AUC when training the logistic regression classifier on the with varying amounts of the dictionary trimmed. Figure 2 plots the learning curves for the training and test set curves for the full dictionary as well as the 20%-80% trimmed dictionary. The training curves show that logistic regression is more sensitive to overtraining, and the reduction of overfitting seen in the 20%-80% dictionary translated into notably improved test error. The gap between the two curves shows that there is still residual overfitting in the final solution, despite the improved score.

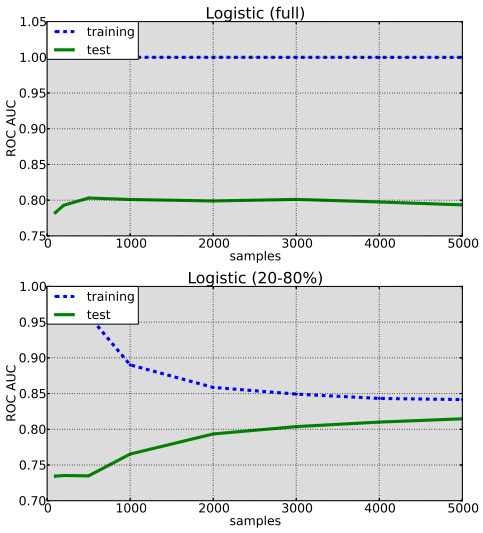

Fig. 2. Learning Curves for Logistic Regression

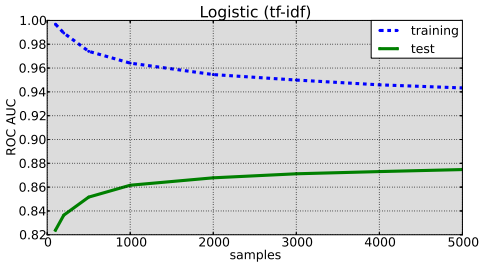

Fig. 3. Learning Curves for Logistic Regression with tf-idf Features

TABLE III. LOGISTIC REGRESSION CROSS-VALIDATION ROC AUC ON TRIMMED DATA

% trim	70	80	90
0	0.823	0.841	0.836
10	0.835	0.834	0.822
20	0.817	0.815	0.811

Since the word frequency dictionary showed that the performance of logistic regression could be improved by reducing overfitting, the use of tf-idf values in the document-word matrix as another approach to feature reduction was investigated next. This approach yielded an improved ROC AUC of 0.875, with the learning curve shown in Figure 3.

Finally, Table III shows the ROC AUC for an SVM classifier operating on document-word frequencies with varying amounts of the dictionary trimmed. For the SVM, trimming high-frequency words had more of an impact than low-frequency ones.

Applying SVM to the tf-idf features resulted in a ROC AUC of 0.819, which was not an improvement on using the trimmed dictionary. Figure 4 shows the learning curve for the 0-80% trimmed dictionary. Note that the SVM does not suffer significantly from the overfitting problem for smaller sample

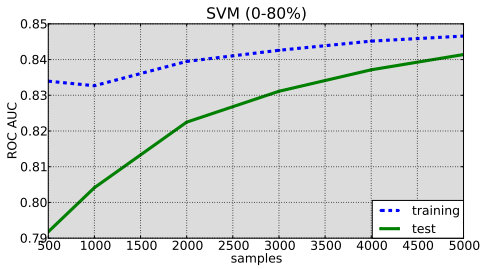

Fig. 4. Learning Curve for SVM

TABLE IV. PREDICTIVE POTENTIAL OF FEATURE SET CANDIDATES

Feature Set	Training ROC	Xval ROC	Precision	Recall
body	0.939	0.875	0.90	0.81
outline	0.920	0.817	0.90	0.73
links	0.904	0.784	0.86	0.86
url	0.910	0.753	0.86	0.87

sizes seen for logistic regression, which likely explains why switching to tf-idf features did not have the same impact here.

B. *Feature Sets*

To evaluate the relative predictive potential of the various feature sets, a logistic regression classifier was trained on each feature set, and the ROC AUC for each classifier was evaluated. The precision and recall for the evergreen classifier based on each feature set was also computed to provide more insight into the nature of the errors of each classifier. Table IV shows the performance of the classifier based on the different feature sets.

The first thing of note is that the text-based features exhibited generally better performance than the url-based features - not surprising, as the text was generally longer and more descriptive (especially the body). Table V shows the top text predictors as identified by the classifier, while Table VI shows the top URL-based predictors. Examination of the word fragments that were most strongly associated with evergreen samples showed topics related to food, health and fitness, and finance. Topics related to technology, fashion, music, and sports tended to be associated with ephemeral samples. The classifiers trained on the url-based features effectively associate different sites with either evergreen or ephemeral content; the "url" feature set, in particular, predicts a sample's class based purely on the site on which it was hosted.

While working with the body text yielded the best results, each classifier was able to correctly predict some samples that others did not. Using N-fold cross-validation, the percentage of samples misclassified by classifier A that were also misclassified by classifier B was estimated. Table VII shows the results.

Since there is a significant subset of errors for all classifiers that are correctly predicted by other classifiers, it seemed likely that training an ensemble classifier using the outputs of the individual classifiers could further improve the score.

TABLE V. TOP EVERGREEN AND EPHEMERAL PREDICTOR WORDS

Top Evergreen Predictors
marthastewart gt signag wing frog delight gina breath tasti crack profil cover pastri halloweekend eco prefer lucyphon tender concentr input meal acn 149 arm hate wheel hazelnut scream cookbook gmo pmid tooth mexican asparagu 2007 hangov leftov ball fondu illustr tension toss hello skin weird tsp teeth random actress yogurt semenya hungri associ kb parchment pet quinoa 2006 tin mouth orang wonder function box cord friend 350 damari salsa submiss ping chick submit gsc crock campi cooksplu text nexeon ricotta way cannot strength artichok coupon seriou cafepress cardiovascular rep lifehack extract dumpl apron african massag heat risk urcrowd hummer dessert direct videoinfo oat granola goat someecard stokk crossbow jar parkour gram clock alyssa tuna sina hydropon veri bite ad salmon 18 ramkissoon espresso period crispi notic chocolat pong decis 30 crunchi peirc feta zest cut rest partnership write technic macaroni elearners phyllo dessertstalk ivillag root formal monsanto stockist bone photographi guru womenswear enjoy brow teaspoon diesel subscrib averag heavi never maegan dinner start sauna sinu method timelin programm soy culinari suggest help herb html5 entertain sesam fred johnni lifestyl mayb margin toothpast databas lasagna three water warrior carbon nuclear diego almond contain broke fit spaghetti omega humorbash chip u00bd sushi pad soft treatment psn hand cracker butterscotch practic tissu boil remov frozen pancak threadless yum stretch dna trainer beat rack tbsp plant pyramid golf shrimp wikihow wall sketch librari milkshak wrote orgasm vegetarian hz melatonin bon cucumb simpl toy indian tabl increas zucchini
Top Ephemeral Predictors
fashion news said he hi iphon ha team smile app price gift easter compani model which video look pumpkin devic dress say show their man boot halloween year that london style on costum sport real imag world product mobil call makeup custom browser de android 2013 system ll googl servic technolog like fall who leagu internet music via trend 2011 than gif hair holiday phone sell fan regist email million hilari america clip govern shirt featur come head rose 3d flight robot stori do wear shoot peopl web back latest outfit cloth qualiti woman bird la see 2012 be print polic digest vaccin 08 fiction view job nike off doe gadget worst basketbal innov 95 offer diamond flu perform insur will melon case report been shoe secur nfl march jean seen inspir anoth street york wood coke event clutch footbal twitter shot social self sunglass softwar humor we script airport color daili facebook game dad geek survey watch usb accessori code must send him china meme cancel screen passov olymp raw break tablet prank mini stay sexi act cute map award friday bracelet here pictur front cloud ill shop allow highlight microsoft polit vehicl leather youtub collect career printer pro credit pay cat mike jersey ladi chic tech patent celebr wors ridicul vodka marijuana star spring everyth amus fame father corpor candi manag soon expens in item crash island where joke hairstyl ndash silk feder hockey believ categori support u00e2 ray ever indi someon machin batteri old church snicker they player boy cost 44 chew 00 rumor lo her court tag announc jami 8230 popul jacket forc sodium suck ski hat decor select jello open cigarett reveal five pretti bra address diabet truffl suffer farmer tenni fli run

TABLE VI. TOP EVERGREEN AND EPHEMERAL PREDICTOR URLS

Top Evergreen Predictors
tammysrecipes.com tandao.com tartelette.blogspot.com tarteletteblog.com tastebook.com tastyhuman.com tedmed.com teknogamia.com tektuff.com templeofthai.com testsounds.com texastypeamom.com thatskinnychickcanbake.com the36thavenue.com thebewitchinkitchen.com thebittenword.com theblackoven.blogspot.com theblackpeppercorn.com theburlapbag.com thecandidappetite.com thecherryblossomgirl.com thecookinghusband.com thecookingphotographer.com thedailyspud.com thedentistauthority.com thefamilyhomestead.com thefightins.com thefrugalgirl.com thegraciouspantry.com thegunnysack.com thegutsygourmet.net thehealthysnacksblog.com thehungrymouse.com theidearoom.net thejewelsofny.com thekindlife.com thelifeofrylie.com thelittleteochew.com themeaningofpie.com therealfoodchannel.com thesneeze.com thespiffycookie.com thestar.co.uk thesweetslife.com thethreecheeses.com thevelvetdoll.com theyummylife.com thisiswhyyourefat.com thismamacooks.com tie timelessinformation.com tiphero.com tonychor.com triathlontrainingschedule.org tulsaworld.com turkishcookbook.com ultimatejujitsu.com unh.edu unitedstatesofmotherhood.com unplggd.com uptowntwirl.com upworthy.com utahdealdiva.com valetmag.com vanillagarlic.com various veganpeace.com vegetarian vintag.es embryo.com visualnews.com vivawoman.net wanderplex.com warpbreach.com warriordash.com washingtonsgreengrocer.com weather.com wechangebeauty.com weddingclan.com weeklycupcake.com weheartfood.com weightlossresources.co.uk weightlosswand.com weightlossweapons.com
Top Ephemeral Predictors
wired.com cool webpronews.com dailymail.co.uk bleacherreport.com popsci.com sports.yahoo.com midwestsportsfans.com itechfuture.com collegehumor.com sportsillustrated.cnn.com vii2012.com cbssports.com forbes.com nydailynews.com technologyreview.com thesun.co.uk techcrunch.com wimp.com globalpost.com splicetoday.com urbanog.com geek.com pcworld.com gawker.com guyism.com thumbpress.com fashionserved.com mymodernmet.com news.com.au pcgerms.com style.com mashable.com nymag.com gadgetrance.com content.usatoday.com mo2no.com pleated technologyinnovationsite.com acecabana.com extremetech.com freep.com newser.com vice.com techflesh.com alternet.org etsy.com polyvore.com threadsence.com washingtonpost.com laweekly.com telegraph.co.uk refinery29.com humor si.com engadget.com rawstory.com reuters.com bucogo.com espn.go.com formyhour.com globalgoodgroup.com kecole.com notalwaysright.com techworldtop.com worldlifeexpectancy.com empowher.com fastcompany.com newseum.org pcmag.com realbeauty.com smbc stumbleupon.com stylelist.com syracuse.com theglobeandmail.com westword.com gizmodo.com cnn.com villagevoice.com mirror.co.uk collegefashion.net cbsnews.com wimp.com break.com upi.com 6jokes.com blog.metmuseum.org blogs.discovermagazine.com dornob.com fashion.elle.com hautemacabre.com icanhascheezburger.com iwastesomuchtime.com moreoo.com newsfeed.time.com politicalhumor.about.com seattletimes.nwsource.com shopruche.com sports.break.com sports.espn.go.com the video.mpora.com

TABLE VII. ERROR CORRELATION OF FEATURE SET CLASSIFIERS

% overlap	B: body	B: outline	B: links	B: url
A: body	100.0	74.3	67.3	58.4
A: outline	51.2	100.0	53.1	49.9
A: links	45.2	51.7	100.0	77.3
A: url	34.3	42.4	67.6	100.0

An ensemble classifier was implemented consisting first of processing the feature sets with individual classifiers, and then constructing a new feature set consisting of the output probabilities from each individual classifier that a given example is in the evergreen class. A regularized logistic regression classifier was then trained on this derived feature set.

The ensemble classifier achieved a K-fold cross-validation ROC AUC of 0.873, with a precision of 0.84 and a recall of 0.80 - effectively identical to the performance of the classifier trained on the body features alone. This was a surprising result in light of the error correlation analysis. One possible explanation is that the ensemble classifier is overfitting the training data: the training ROC AUC is much higher, at 0.954. Some evidence for this can be seen from the fact that when the regularization constant was tuned empirically for the ensemble classifier, the optimal value called for a significantly higher penalty coefficient than the classifiers for the individual feature sets. This suggests that obtaining additional samples to reduce the overfitting could result in better performance for the ensemble classifier.

C. Error Analysis

Table IV showed that the leading classifier, logistic regression on the body text features, shows somewhat better precision than recall, indicating that errors are slightly biased towards false negatives in the evergreen class.

To better understand the nature of the classification errors, an investigation of some of the misclassified examples was conducted. This turned up the interesting result that the labels in the training set themselves are fairly noisy. For example,

the following two URLs were two different training examples in the dataset:

- http://news.menshealth.com/touch-at-your-own-peril/2011/11/02?cm_mmc=Facebook-_-MensHealth-_-Content-MHNews-_-6DirtiestPlaces
- http://news.menshealth.com/touch-at-your-own-peril/2011/11/02/

The URLs actually referred to the same page; the first link simply contained an extra query string parameter (likely representing the referring page) which did not affect the retrieved content. However, the first URL was labeled as ephemeral, while the second was labeled as evergreen. There were numerous other examples of effectively identical or similar pages that had conflicting labels. Since the examples were hand-labeled, this likely represents either human error or a difference of opinion on the part of the humans performing the labeling.

A systemic analysis of the URLs shows that 1.8% samples have the same URL and differ only in the query string, but are classified differently. Spot-checking several such URLs suggests that they are most likely misclassifications of the same sort as the menshealth.com example. There are also additional cases that are somewhat more ambiguous but also likely misclassifications: for example, two receipes for different dishes hosted on the same domain. Since it is difficult to distinguish these cases from pages hosted on the same domain that are legitimately different classes, no attempt has been made to quantify these cases, but it suggest that the 1.8% figure is likely a lower bound.

The misclassified examples affects the accuracy of the classifier since they add noise to the input. They also affect the ROC AUC estimate, since examples which the classifier correctly labels but which for which the training label is incorrect will be mistakenly considered to be in error. Increasing the number of training examples gathered would help address both these problems.

IV. Conclusions and Future Work

Several popular classification algorithms were implemented and applied to the evergreen classification problem. From the results, we observe that logistic regression is well suited to this text classification problem, achieving the best ROC AUC score of the classifiers investigated.

Feature extraction for the classification problem was examined, and low-content features such as javascript and portions of the HTML markup were identified and removed in preprocessing. Salient features such as the body text, document outline, and relevant links/urls were extracted and their predictive power analyzed by training independent classifiers on them. The body text and document outlines were more successful than the link-based features, with predictions based on just the outline able to achieve a ROC AUC within 6 percentage points of the full body text. Tf-idf proved to be effective in further extracting the most relevant terms within the text-based features.

While a comparison of the misclassified samples from classifiers trained on the different feature sets suggested that somem benefit could be achieved by combining the information from the separate predictions, attempts do so via an ensemble classifier did not significantly improve the result. Training curve analysis showed a gap in convergence between training and test ROC AUC, indicating that the ensemble classifier is overfitting. Error analysis also indicated that the training set itself is subject to mislabeling, introducing a further source of noise in the dataset.

For both these reasons, a fruitful area of future work is likely to gather additional examples to expand the training set and reduce the overfitting. This will assist with overfitting, as well as to help compensate for mislabeling errors. The ensemble classifier should be revisited in this context, as the analysis of error corrleation and learning curves suggested that it could improve prediction if the overfitting issues were addressed.

A second avenue for investigation is using alternatives to hand-labeling the samples, since the hand-labeling is dependent on the opinions of a small number of human labelers. If StumbleUpon is able to add tracking of click-throughs to articles presented to real end users, the tracking data could be used to establish the longevity of the content in question more directly, which should help reduce mislabeling.

References

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

[2] T. Fawcett, “An Introduction to ROC Analysis”, *Pattern Recognition Letters*, Issue 27, 2006, pp 861-874

[3] Wikipedia, “Receiver Operating Characteristic”, http://en.wikipedia.org/wiki/Receiver_operating_characteristic

[4] M.F. Porter, “An algorithm for suffix stripping”, *Program*, 1980, pp 130-137

[5] Wikipedia, “Bag-of-words model”, http://en.wikipedia.org/wiki/Bag-of-words_model

[6] Wikipedia, “tf-idf”, http://en.wikipedia.org/wiki/Tf%E2%80%93idf