Forecasting aggregate reactions to blog-posts

Jigyasa Grover
University of California, San Diego
9500 Gilman Dr, La Jolla
CA 92093 USA
jigrover@ucsd.edu

Chand Anand
University of California, San Diego
9500 Gilman Dr, La Jolla
CA 92093 USA
canand@ucsd.edu

ABSTRACT

Online social media has impressively facilitated creation and sharing of information, ideas, interests and other forms of expression via myriad virtual communities and networks. In this bracket, blogs consisting of discrete, often informal entries ("posts") dispense functional means to gauge trending topics and user interests over the internet. This work focuses on estimating the reactions a blog-post might receive in the next chosen interval of time in the form of comments by users based on chief attributes of the blog-post by deploying state-of-the-art machine learning regression techniques.

Keywords

Social Media; Blog-Posts; Comments; Machine Learning; Regression; Lasso Regression; k-NN; Decision Trees; Random Forest; Bagging; Gradient Boosting; Multilayer Perceptrons (MLP)

1. INTRODUCTION

In this 21st century, online social media has become the platform of choice for user interactions over the internet. Initially explored for rudimentary purposes like reading articles, sending messages or watching movies, internet particularly social media has sprouted enormously providing means to create, share, tag content and converse creatively. It has now come across as a multifaceted platform serving entertainment, news, classifieds, product advertisements, etc. consequently dominating the way how people accumulate and perceive information in today's world of temporal complexities. Thus, fathoming this contemporary social nexus gives valuable insight into human interactions and social psychology.

One of the compelling tasks is analysis of the content posted in the rich ecology of internet in form of blog-posts and especially their feedback and reactions to comprehend real-world observations of user behaviour incorporating all elements like trending topics, instant of publishing, verbosity etc. Undoubtedly, human experts are incompetent to perform this detailed analysis without deploying smart systems to do the required. Hence, dynamically evolving and scalable techniques are looked forward to in this era of multiplexity along with the flexibility to mould the source code to delve into huge amount of raw data from different perspectives and lead to interesting predictions and conclusions by examining such social media documents which are undirected, voluminous and alternating.

In this study, we focused on forecasting the aggregate reactions a blog-post might receive in the form of comments in the next chosen interval of time. We took into account the documents appearing in Hungarian blog-posts crawled as a part of a similar experiment *Feedback Prediction for Blogs* by *Krisztian Buza* [1].

Here, Section 2 presents a brief summary of works aligned with our study comprising literature survey and realm review. In Section 3 and Section 4, domain-specific concepts and prime insights into the data-set being used are put forth respectively. Section 5 formulates the predictive task undertaken, defines model evaluation metric and feature extraction process. We then enlist the models deployed their implementation and failed attempts in Section 6. Finally, Section 7 sums up the work by presenting the result analysis and inferences thus deduced.

2. RELATED WORK

Online social media content of diverse nature is now available effortlessly owing to exhaustive research on data mining techniques, refer Adedoyin-Olowe et. al [2] with special emphasis on blog data mining in the works of Dou Shen et. al [3]. There have been improvements in these data extraction techniques by deploying innovative opinion extraction and topic tracking approaches as done by Khan et. al [4] which used opinion lexicons & user defined lexical dictionaries for semantic orientation. Chen et. al [5] also inspire with their work on topic tracking for micro-blog texts based on semantic relevance.

The theme of predicting the aggregate reactions to a blog-post in the form of comments is researched upon atypically. Mishne and Glance [6] analyzed the co-relation between popularity of a blog and the commenting patterns on it using sizable corpus of comments from the blog-o-sphere. It thus exhibited the value of comments in characterizing the social repercussions of a post, including popularity and controversy. [7]

Likewise, Yano et. al [8] modeled the generation of the primary documents (online political blog-posts of Matthew Yglesias [9] and RedState [10]), authorship, and contents of the blog community's verbal reactions to each post in the form of comments by extending Latent Dirichlet Allocation, introduced by Blei et al. [11]. The model was then evaluated on a novel comment prediction task which estimated which readers were likely to leave comments on the given post.

The most closely related work of Yano et. al [7] models the relationship between the text of a political blog-post (The Atlantic - Matthew Yglesias [9] and RedState [10]) and the volume of comments received by a post which was further

generalized by Krisztian Buza [1] to predict the number of comments that a blog-post is expected to receive in the next 24 hours by using Multilayer Perceptrons (MLP), linear regressors, RBF-Networks, REP-Trees and M5P-Trees.

In resemblance and contrast of the above, we have performed experiments with varied models including Lasso Regression, k-Nearest Neighbors (k-NN) Regression, Multilayer Perceptrons (MLP) learning algorithm, Decision Tree Regression, Random Decision Forest and Gradient Boosting after doing sizeable pre-processing on the corpus of Hungarian blog-posts encompassing multitude of topics to predict the number of comments in the next $H = 24$ hours. It can be discerned from our work that the conclusions align with the related works in this domain.

3. DOMAIN-SPECIFIC CONCEPTS

Some domain-specific concepts have been described below in order to assist lucid addressing of the problem. A *source* is said to produce *documents*. As an example, on the website *myblog.blog.hu*, new documents appear regularly, therefore, *myblog.blog.hu* is the source of these documents.

We have considered the documents to be made up of the following components in our work:

- *main text of the document:* the text that is written by the author of the document, this text describes the topic of the document,
- *links to other documents:* pointers to semantically related documents, in our case, trackbacks are regarded as such links,
- *feedback:* opinions of social media users about a document is very often expressed in form of feedback that the document receives. Feedback are usually short textual comments referring to the main text of the document and/or other feedback.

As we were looking through the temporal aspects of the above entities, e.g. taking into account the comment volume history 48 to 24 hours before the *basetime*, the feedback and thier time-stamps were extracted correspondingly.

4. DATASET INSIGHT

The selected dataset [12] from University of California - Irvine's machine learning repository originates from a collection of 60,021 Hungarian blog-posts with 37,279 raw HTML-pages crawled and processed from roughly 1,200 sources. The task is to predict the number of comments in the next chosen interval of time, i.e. $H = 24$ hours. Hence, a *basetime* was chosen in the past and the blog-posts that were published at most 72 hours before the selected base date/time were selected to simulate this situation. Thereafter, the feature set of the selected blog-posts from data available at *basetime* was assembled.

The selected dataset contains following features for each blog-post that appeared has in the past:

1. **Basic features**: Number of links and comments in the previous 24 hours relative to *basetime*; number of links and comments in the time interval from 48 hours prior to *basetime* to 24 hours prior to *basetime*; how the number of links and comments increased/decreased in the past (the past is seen relative to *basetime*); number of links and comments in the first 24 hours after the publication of the document, but before *basetime*; aggregation of the above features by source.
2. **Textual features**: Bag of words features for the 200 frequent words of the text of the blog-post.
3. **Weekday features**: Binary indicator features that describe on which day of the week the main text of the document was published and for which day of the week the prediction has to be made.

Table 1: Feature Information

Feature	Type	Count
Basic		
Average, standard deviation, min, max and median of the next 5 attributes for the source	float	50
Total # comments before basetime	int	1
# comments in the last 24 hours before basetime	int	1
# comments in between 48 hours and 24 hours before basetime	int	1
# comments in first 24 hours after publication and before basetime	int	1
Difference between # comments in 48-24 hrs and last 24 hrs before basetime	int	1
Total # links before basetime	int	1
# links in the last 24 hours before basetime	int	1
# links in between 48 hours and 24 hours before basetime	int	1
# links in first 24 hours after publication and before basetime	int	1
Difference between # of links in 48-24 hrs and last 24 hrs before basetime	int	1
Length of time between the publication of the blog-post and basetime	int	1
Length of the blog-post	int	1
Textual		
Bag of words features for 200 frequent words of the text	int	200
Weekday		
Weekday (Monday...Sunday) of the basetime	boolean	7
Weekday (Monday...Sunday) of the date of publication	boolean	7
Parent		
# parents	int	1
Minimum, maximum, average # comments that parents received	float	3

4. **Parent features**: A document P_d is a parent of document d, if d is a reply to P_d, i.e., there is a trackback link on P_d that points to d; parent features are the number of parents, minimum, maximum and average number of comments that the parents received.

We have a total of 60,021 instances of blog-posts, each comprising 281 attributes (including 1 target field). In the train data, the *basetimes* were in the years 2010 and 2011. In the test data the *basetimes* were in February 2012 and March 2012. This simulates the real-world situation in which training data from the past is available to predict events in the future.

Figure 1: Histograms of Target No. of comments

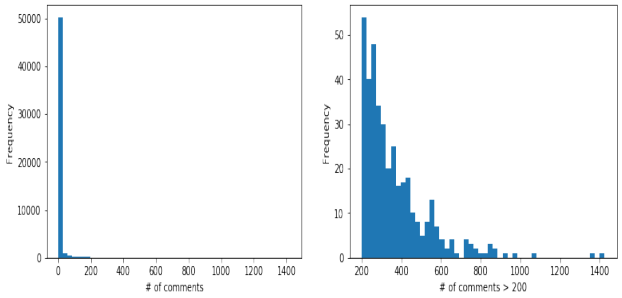
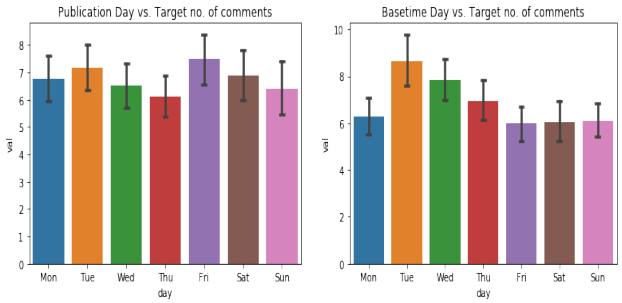


Figure 2: Trend with days of week



As can be seen from the Figure 1, the target values are highly skewed. In order to mediate the skewness of the data, we scale the data and transform using log. From Figure 2, we can infer that there is no significant effect of the days of the weeks on target number of comments. We also observed that certain attributes' value had no variation, hence providing no additional information for our model construction. Also, there were many instances where the length of blog-post was less than 100 words, we dropped such instances assuming erroneous or trivial content. There were also cases of many outliers in the dataset with target number of comments more than 1000. Thus, we performed pre-processing of the dataset before proceeding.

Table 2: Target number of comments statistics

count	52397.000000
mean	6.764719
std	37.706565
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1424.000000

5. PREDICTIVE TASK

The task here is to estimate the number of comments a *recently* published blog-post tends to receive in the next H hours, based upon the prior knowledge of blog-posts published in the past along with the number of reactions they received. In our case study, we have assumed that a blog post is said to be published *recently* if done in last 72 hours and we are setting H to be 24.

To address the problem in question, we brought into service machine learning regression models where the recently published blog documents were treated as *instances* and number of feedback that the blog-post will receive in the next H hours as the *target*. While constructing the regressor, our prediction model considered the training dataset where the *target*, or the number of comments received by the blog-post was known. The regression model thus constructed was put into application on the testing dataset to perform quantitative evaluation.

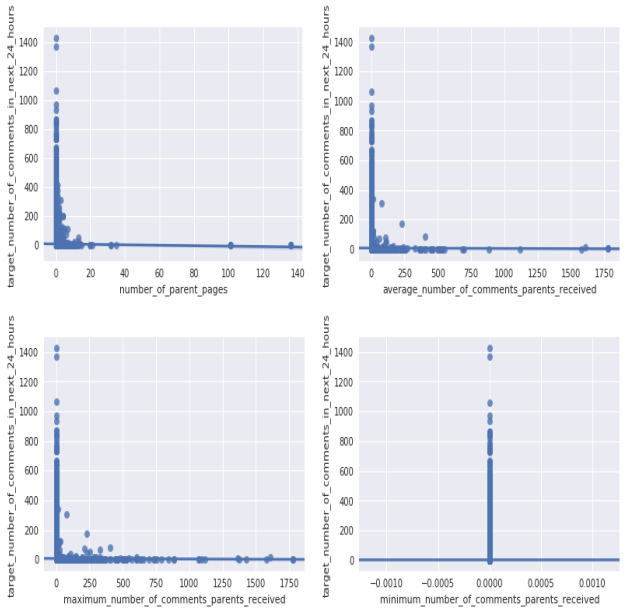
5.1 Model Evaluation

The evaluation of the model is one of the principal tasks in a machine learning project which delineates how good the predictions made by the model are. From a huge variety of handy performance metrics available, we examined the performance of all our regression models based on *Mean Squared Error (MSE)*. It was presumed that a model with lower MSE performed better.

Mean Squared Error (MSE).

Mean-square error (MSE) is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values

Figure 3: Parent features



actually observed. MSE has proven to be useful in relaying the concepts of bias, precision, and accuracy in statistical estimation. The MSE of predicted values $\hat{y}_i$ for ith observation of a regression's dependent variable y_i is computed for n different predictions as the mean of the squares of the deviations:

$MeanSquaredError(MSE) = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}$

5.2 Feature Extraction

As can be observed from Table 1, the dataset being used incorporates several kinds of characteristic attributes namely, basic features (i.e. comment statistics), textual features (i.e. bag of words features), weekday features (i.e. temporal information) and parent features (i.e. information pertaining to parent of the posts). Slight investigation of the dataset as done previously points that weekday features and parent features do not provide additional direction to our model construction. For that reason, to have a finer analysis we build three different sub-datasets to work on:

- Basic: Consisting of basic features only
- Basic + Textual: Consisting of basic and textual features
- Textual: Consisting of textual features only

The sub-datasets being used spanned considerable dimensions. In order to have dense information in less number of dimensions, we applied dimensionality reduction techniques. For our work, we reduced each of the sub-datasets' features using two techniques as follows:

- Orthogonal transformation of a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables using *Principal Component Analysis (PCA)*
- Important Feature Extraction using randomized decision trees via *Extra Trees Regressor (ETR)*

6. MODELS DEPLOYED

We aimed at selecting a representative set of state-of-the-art machine learning regression models for our experiment encompassing: Lasso Regression, k-Nearest Neighbors (k-NN) Regression, Multilayer Perceptrons (MLP) Learning, Decision Tree Regression, Random Decision Forest and Gradient Boosting. We have used the *scikit-learn* [13] implementations of these regression models based on varied theoretical backgrounds of machine learning concepts. 10-fold cross-validation technique was used to tune the parameters of the aforementioned models.

The selected models are appropriate for the type of prediction task that we are undertaking as each of the model deployed is designed to estimate probabilities thus assisting tremendously in predictive task like ours. The models thereby used are sufficiently different in their approaches to suggest that there will be variance in how well they perform.

6.1 Baseline

Predominantly, baseline method uses simple summary statistics and heuristics to create predictions for a dataset which then becomes a metric to compare any other machine learning algorithm against. Here, we are using the average number of comments calculated from the training dataset as the target estimated number of comments a blog-post tends to receive in the next $H = 24$ hours. We obtain a mean squared error of 1.3684 on the above baseline after dimensionality reduction using PCA and ETR. All the other successful models are looked forward to outperform this baseline score.

6.2 Lasso Regression

Lasso (least absolute shrinkage and selection operator) Regression performs both variable selection and regularization in order to enhance the prediction accuracy. In simpler terms, performing linear regression with L1 prior as regularizer is Lasso Regression. By default, 1000 iterations were performed with 0.0001 as the tolerance for optimization where coefficients were updated by looping over features sequentially. Though Lasso regression makes a model interpretable by performing fine subset selection, it retains only one of the correlated variables setting the rest correlated variables to zero thus possibly leading to some loss of information resulting in lower accuracy in our model.

6.3 k-Nearest Neighbors (k-NN) Regression

k-nearest neighbors algorithm (k-NN) outputs the average of the values of its k nearest neighbors in a non-parametric fashion when exploited for regression tasks. k-NN is robust to fluctuations in feature values by taking an aggregate similarity measurement across features (L2-Norm). The impediments faced in using k-NN algorithm are the high computation costs, dependent similarity measures (for example, distance measure) and evaluation of optimal k. We performed a search over a list of multiple k values and then picked the one with the lowest MSE. After experimentation, we got the optimal value of k to be 3 for which MSE was around 0.8436 for our basic sub-dataset.

6.4 Multilayer Perceptrons (MLP) Learning Algorithm

Multilayer perceptron (MLP) is a class of feed-forward artificial neural network that learns a non-linear function approximator for regression tasks where the perceptrons act as hidden feature extractors. Adaptive learning where no major assumption is made regarding the underlying probabilistic information of the data thereby proving to be a *universal approximator* forms the core strength of MLP. However, if the the number of layers and neurons set by the user are not optimal, it might lead to drastic under-fitting or over-fitting of the model. There might be even cases where the algorithm gets stuck in an infinite loop in the local minima consequently inferring it to be a global one. In our model construction, we analytically used 2 hidden layers with 10 and 5 neurons each for our model which optimized the squared-loss using stochastic gradient descent.

6.5 Decision Tree Regression

Decision Tree builds regression models in the form of a tree structure using ID3 algorithm [14] employing a top-down, greedy search through the space of possible branches with no backtracking. Decision Trees implicitly perform variable screening without letting the non-linear relationships between parameters affect its performance. However, Decision Trees tend to be extremely sensitive to small perturbations in the data and can easily over-fit if validation methods and pruning are neglected. In our experiment, we considered all features while looking for the best split and MSE was used

as the function to measure the quality of a split where the nodes were expanded until all leaves were pure or until all leaves contained less than $min\_samples\_split = 2$ samples.

6.6 Random Decision Forest

Random Decision Forest is an ensemble learning method which constructs a multitude of Decision Trees. It is thus able to learn non-linear hypothesis and gives a very good generalization performance and does not over-fit data. Conversely, for data including categorical variables with different number of levels, random forest is biased in favor of attributes with more levels thus rendering the model unreliable in those cases. For our experiment, we used 10 trees and the maximum number of features considered at each decision node was $log_2(total\_features)$. We used the default *Gini impurity* [15] as the function to measure the quality of a split where the nodes of the regression tree were expanded until all leaves were pure or until all leaves contained less than $min\_samples\_split = 2$ samples. As anticipated, it had one of the top performances for our predictive task.

6.7 Gradient Boosting

Gradient Boosting builds an additive model in a forward stage-wise fashion and it generalizes them by allowing optimization of an arbitrary differentiable loss function. In each of the 100 boosting stages performed, regression trees were fitted on the deviance loss function, where we set maximum depth of the individual regression estimators as 3. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance. Though it requires careful tuning of the model parameters and fairs poorly while extrapolating.

6.8 Meta-Algorithms

To obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone, we used bagging and boosting meta-heuristics designed to improve solution to an optimization problem, especially for limited computation capacity and/or imperfect information.

6.8.1 Bagging

Bagging (**B**ootstrap **agg**regat**ing**) is a machine learning ensemble meta-algorithm originally proposed [16] to improve the stability and accuracy of machine learning algorithms. We applied the model averaging approach of the bagging technique to Random Forest and Gradient Boosting methods to reduce variance and avoid over-fitting.

6.8.2 Boosting

To reduce bias and variance in our regression models, we used boosting ensemble meta-algorithm which improved our results impactfully. Generally, boosting transfigures a set of weak learners to that of a strong one where a weak learner is a classifier which is only slightly correlated with the true classification and a strong learner is a classifier that is arbitrarily well-correlated with the true classification. We performed boosting by giving 0.25 and 0.75 weights to Random Forest and Gradient Boosting respectively.

Failed Attempts.

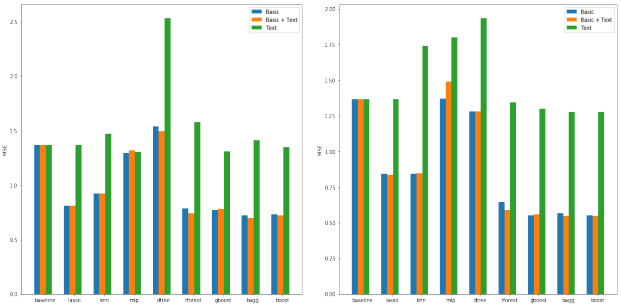
During the course of our regression model constructions and implementation, we came across certain scenarios which did not furnish good results in comparison thus giving us a better insight into the problem and helped us build more sturdy models. We observed that individual regression models namely Lasso Regression, k-NN Regression, MLP Regression and Decision Tree Regression did not perform remarkably well encouraging us to use ensemble learning methods for Gradient Learning (Gradient Boosting) and Rule-based Learning (Random Forest).

7. RESULT ANALYSIS & INFERENCES

Our brief analysis of estimating the number of comments a blog post is inclined to receive from all its readers in the next $H = 24$ hours using state-of-the-art machine learning regression models leads us to results as shown in Table 3.

As can be observed from the performance of the regression models for different features sets viz. PCA and ETR, the trend of MSE variation is similar for Basic Features, Basic + Textual Features and Textual Features. Demonstrably, Basic + Textual features give a better performance owing to their supply of key attributes of the blog-post as opposed to only Textual or only Basic features.

Figure 4: MSE on PCA (l) & ETR (r) Feature Set



While using PCA dataset, Bagging yielded the best result with 0.6990 mean squared error followed by 0.7247 of Boosting thus both outperforming baseline substantially. Ensemble learning methods Random Decision Forest and Gradient Boosting too gave a low MSE of 0.7435 and 0.7824 respectively in comparison to baseline score.

Similar tendencies were observed while using the reduced ETR feature set where Boosting performed the best delivering a MSE of 0.5471 followed closely by Boosting with an MSE of 0.5478.

It can thus be noticed that ensemble learning methods, Random Decision Forest and Gradient Boosting outperform the baselines and other regression methodologies namely Lasso regression and k-NN Regression because of being inherently robust to over-fitting thereby lending themselves well to our dataset. Further, as can be observed from Figure 4 employing modern techniques of bagging and boosting lead to reduced variance and increased stability of the predictive model constructed in both cases of PCA and ETR feature sets.

8. ACKNOWLEDGMENTS

Due thanks to Prof. Julian McAuley (Instructor, UCSD CSE 258: Web Mining & Recommender Systems - Fall 2017) and all the teaching assistants for their continuous support and guidance.

Table 3: Performance (MSE) of the regression models for different feature sets

	PCA			***ETR***		
	Basic	**Basic + Textual**	**Textual**	**Basic**	**Basic + Textual**	**Textual**
Baseline	1.3684	1.3684	1.3684	1.3684	1.3684	1.3684
Lasso Regression	0.8088	0.8088	1.3684	0.8435	0.8351	1.3684
k-NN Regression	0.9228	0.9228	1.4698	0.8436	0.8479	1.7400
MLP Regression	1.2948	1.3192	1.3040	1.3700	1.4890	1.8000
Decision Tree	1.5381	1.4943	2.5327	1.2806	1.2819	1.9343
Random Forest	0.7878	0.7435	1.5809	0.6455	0.5884	1.3456
Gradient Boosting	0.7738	0.7824	1.3086	0.5537	0.5593	1.3010
Bagging	0.7223	0.6990	1.4113	0.5679	**0.5478**	1.2775
Boosting	0.7334	0.7247	1.3516	0.5529	**0.5471**	1.2778

9. REFERENCES

[1] Krisztian Buza. Feedback Prediction for Blogs, 2014.

[2] Mariam Adedoyin-Olowe, Mohamed Medhat Gaber, and Frederic T. Stahl. A Survey of Data Mining Techniques for Social Media Analysis. *CoRR*, abs/1312.4617, 2013.

[3] D. Shen, J. t. Sun, Q. Yang, and Z. Chen. Latent Friend Mining from Blog Data. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 552–561, Dec 2006.

[4] Aurangzeb Khan, Khairullah Khan, Shakeel Ahmad, Fazal Masood Kundi, Irum Tareen, and Muhammad Zubair Asghar. Lexical Based Semantic Orientation of Online Customer Reviews and Blogs. *CoRR*, abs/1607.02355, 2016.

[5] H. Chen, J. Lu, F. Wang, Y. Zhang, and S. Zhao. A New Method of Topic Tracking for Micro-Blog Texts Based on Semantic Relevance. In *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 2, pages 349–353, Aug 2017.

[6] Gilad Mishne and Natalie Glance. Leave a Reply: An Analysis of Weblog Comments. In *Third annual workshop on the Weblogging ecosystem*, 2006.

[7] Tae Yano and Noah A. Smith. What's Worthy of Comment? Content and Comment Volume in Political Blogs, 2010.

[8] Tae Yano, William W. Cohen, and Noah A. Smith. Predicting Response to Political Blog Posts with Topic Models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 477–485, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[9] Matthew Yglesias. All Stories by Matthew Yglesias on The Atlantic, 2007.

[10] www.redstate.com. RedState: Conservative Blog & Conservative News Source for Right of Wing Views, 2007.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[12] UCI Machine Learning Repository: Center for Machine Learning and Intelligent Systems. Blog-Feedback Data Set, 2014.

[13] David Cournapeau. scikit-learn: Machine Learning in Python, 2007.

[14] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, March 1986.

[15] Antonio D'Ambrosio and Valerio A. Tutore. *Conditional Classification Trees by Weighting the Gini Impurity Measure*, pages 273–280. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[16] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.