Comment Classification for an Online News Domain

Dirk Brand
Computer Science Division
Stellenbosch University
7602 Matieland
South Africa
dirkbrand@ml.sun.ac.za

Brink vd Merwe
Computer Science Division
Stellenbosch University
7602 Matieland
South Africa
abvdm@cs.sun.ac.za

ABSTRACT

In online discussion forums, comment moderation systems are often faced with the problem of establishing the value of an unseen online comment. By knowing the value of comments, the system is empowered to establish rank and to enhance the user experience. It is also useful for identifying malicious users that consistently show behaviour that is detrimental to the community.

In this paper, we investigate and evaluate various machine learning techniques for automatic comment scoring. We derive a set of features that aim to capture various comment quality metrics (like relevance, informativeness and spelling) and compare it to content-based features. We investigate the correlation of these features against the community popularity of the comments. Through investigation of supervised learning techniques, we show that content-based features better serves as a predictor of popularity, while quality-based features are better suited for predicting user engagement. We also evaluate how well our classifier based rankings correlate to community preference.

General Terms

Algorithms, Experimentation

Keywords

Regression, Classification, Features

1. INTRODUCTION

There are various online platforms that permit users to generate content. These include forums, blogs, newsgroups and online news providers. The content often has to be moderated for public and corporate benefit. Moderation in the online news domain has recently been a topic of discussion, as users are ever more able to voice their opinions about reported news via some social platform. As the social web grows and people become increasingly socially aware, news sites are becoming ever larger discussion communities where users can address and comment on common issues spurred by the news articles [1]. One of the key features promoting the success of these online communities is the large-scale user-engagement, seen in the forms of rating, tagging and commenting on content [2]. User-contributed comments offer a much richer source of contextual information than ratings or tags, albeit often a “messy” source of information. Comments are often variable in quality, substance, relevance and style [2].

An online news portal serves many different roles [3]. These rolls fulfil the following tasks:

- educating people,
- providing instant access to the latest news,
- providing feedback for news provider, and
- easily accessible source of information for the general public.

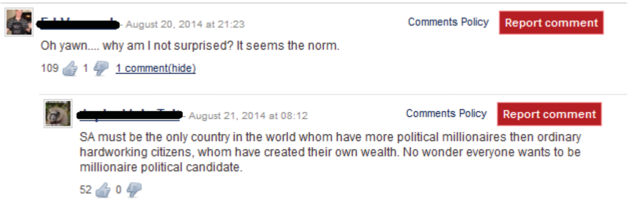
The importance of the roll that online news play in the media sector (especially when educating and informing people) leads news providers to strive to provide content of higher quality. To ensure high quality in user submitted content (such as comments on articles), news providers attempt to moderate or curate the content. Several systems of content moderation have been designed and implemented in the past. These will be explained in Section 7.

2. PROPOSED APPROACH

Previous studies [4, 5, 2] have investigated classification techniques and regression approaches for ranking comments. The authors mentioned above extracted quality-based features from comments using some of the feature extraction techniques mentioned in Section 3. We attempt to show that the same quality-based features are insufficient for predicting a comment’s popularity within the community, but that using only content-based features are better suited (or can at least serve to augment traditional quality-based features). We will also compare the efficiency of the two feature sets for predicting user engagement.

For the quality-based features, we incorporate feature extraction techniques from previous authors. The content-based feature extraction is a new technique for comment ranking (to our knowledge) and seeks to improve on the proposed techniques by instead using a bag-of-words vectors as a feature set. We predict that the quality-based features might be a better predictor of editor preference (over community preference), but the provided data was insufficient to test this hypothesis.

For the supervised learning approaches, a regression filter [6, 7] is applied to a comment and it classifies the comment based on a provided feature set. The regression classifier predicts a continuous numerical value for a comment. We will also investigate the effect of categorising the dependent variable and translating the problem into a classification problem.



[redacted] - August 20, 2014 at 21:23

Oh yawn.... why am I not surprised? It seems the norm.

109 👍 1 👎 1 comment(hide)

[redacted] - August 21, 2014 at 08:12

SA must be the only country in the world whom have more political millionaires then ordinary hardworking citizens, whom have created their own wealth. No wonder everyone wants to be millionaire political candidate.

52 👍 0 👎

Figure 1: A typical News24 comment thread.

The features will all be extracted from a comment database provided by News24.com (the nature of the data sets are explained in Section 6.1). News24 is a popular South African news provider that allows its users to leave comments on articles. Figure 1 shows a typical comment thread where one user posted a comment and another user commented on his comment (this is an example of the 1-tier commenting that News24 permits). Each article has multiple threads of comments associated with it. These comments will form the basis for our investigation.

3. FEATURE EXTRACTION

Training data is comprised of rows that each consist of a feature vector and an associated value. The model is trained on this training set. The model can then be used to predict the value of a candidate feature vector (a new comment, for instance). The choice of features to use in the training data depends on the domain of the data, as well as the relevance of the features [8].

Consider a set of articles $\{a_1, a_2, ..., a_k\}$. Denote the i_{th} article by a_i, and its set of n comments by $\{c_{i1}, c_{i2}, ..., c_{in}\}$. For each comment c_{ij}, a set of features $F_{c_{ij}} = \{f_1, f_2, ..., f_m\}$ is extracted. The training data then consists of rows of the form $\{(F_{c_{11}}, r_{c_{11}}), ..., (F_{c_{kn}}, r_{c_{kn}})\}$ where a tuple $(F_{c_{ij}}, r_{c_{ij}})$ indicates a feature set $F_{c_{ij}}$ for comment c_{ij}, and the associated community rating $r_{c_{ij}}$.

We extracted quality-based features, based on previous work. We then explain how a content-based feature set is extracted as a comparison.

3.1 Quality-Based Features

The various features used for the quality-based feature set, are discussed below. The features can be categorised into surface features, lexical features and sentiment features.

- **Surface Features**
  - *Timeliness.* This feature reflects the response time of a user's comment in relation to when the relevant article was posted [9].
  - *Lengthiness.* This feature is a simple measure of the length of a comment relative to the average length of comments of that article [9].
  - *Uppercase Frequency.* This feature is a count of the number of words that are completely uppercase [2].
  - *Question and Exclamation Frequency.* Both features are the counts of the number of sentences in the comment that end in question and exclamation marks respectively [10]. The values are given as a percentage of the total number of sentences.
- **Lexical Features**
  - *Complexity.* The complexity of a comment is measured by the entropy of the words in the comment [2]. Intuitively, it represents the diversity in word choice in the comment. A low entropy score would indicate that a comment has few or repetitive words.
  - *Spelling.* This feature measures the frequency of misspelled words in the comment. The feature is calculated by looking up each word in a dictionary and recording the percentage of words that cannot be found in the dictionary. The dictionary is comprised of words extracted from Peter Norvig's spell checker data sources [11] and the NLTK [12] sources for male and female names. In future, it would be beneficial to collect data on South African English spelling and use that for the feature.
  - *Profanity.* This feature measures the frequency of profane words in the comment. Similar to the spelling feature, the feature value is calculated by looking up each word in a dictionary of profane language and recording the percentage of words that can be found in the list of banned words. The list is built from a list published by Alejandro U. Alvarez [13][1].
  - *Informativeness.* This feature attempts to capture how unique a comment is within its relative thread. The measure that was used, is the standard Term Frequency - Inverse Document Frequency (TF-IDF [14]).
  - *Readability.* The readability of a comment is defined as with what ease the reader is able to read the comment (determined by the Flesch Reading Ease Test (FRES) [15]). A high score (above 90) indicates that the text can be understood by an average 11-year old, whereas conversely, a low score (between 0 and 30) indicates that the text will probably only be understood by university graduates.
  - *Relevance.* The relevance of a comment can be measured relative to the article or relative to the comment thread that the comment is present in. To calculate the relevance within the comment thread, the overlap between the words in the comment and the words in the article's comments, is quantified. For this, a bag-of-words vector of the 100 most frequent words is generated from all the

[1] The list tries to take common purposeful misspellings of words into account. Eg. 'butt' and 'buttt' are both in the list. In future, a more domain specific list should be constructed.

comments on an article. Similarly, to calculate a comment's relevance to the article, a bag of words is generated from the body of the article.

- **Social Features**
  - *Sentiment.* The text in a comment can be classified as either subjective or objective, and further more as positive or negative (if it was classified as subjective). A trained classifier was used to predict the sentiment of a comment. The classifier was trained and tested with a corpus of 100,000 real tweets (from Twitter [2]) that were classified as either positive or negative. The classifier achieved a prediction accuracy of 84.7%.
  - *Subjectivity.* The subjectivity of the comment is also captured as a feature. If a comment is between 45% and 55% positive, the comment is classified as objective, otherwise it is classified as subjective.
  - *Engagement.* Since News24 uses a one-tier commenting system, users can either leave a new comment ("parent" comment) or comment on an already posted comment ("child" comment). This feature counts the number of child comments to each parent comment.

3.2 Content-Based Features

The above mentioned features attempt to capture the "quality" of a comment. Another way to characterise a comment, is to use the actual content of the comment. To capture this, a list of the most used words in the entire comment space is compiled. Then, for each comment, a vector of the number of occurrences of each word in the comment is created.

For accuracy, stopwords [16] are not consider for the frequent words list. Also, only the stems of words are considered. This is done to group plurals and other word variations into a single representative stem. The Porter stemming algorithm [17] is used (from within the NLTK package [12]) for stemming.

3.3 Value Extraction

The supervised learning methods require a dependent variable (or predictor). Two measures of determining the dependent variable are investigated for this project, engagement and popularity. The engagement that a comment attracts is measured in the percentage of votes on an article, while the popularity is measured by the vote ratio. For the classification methods, the values are discretised into two, three or five balanced categories. The details of the methods are:

- **Percentage of Total Votes** - The ratio of likes to dislikes of a comment: $v = (likes{+}dislikes)/(\#article\ votes)$.
- **Vote Ratio** - The ratio of likes to dislikes of a comment: $v = likes + c/(likes\ +\ dislikes\ +\ 2 * c)$. c is a correction term to deal with comments with zero likes or dislikes (set to 5 in our experiments).

4. DATA PREPROCESSING

Various assumptions are made about the training data for the regression and classification models [18]. Firstly, regression (specifically) assumes that each feature is normally distributed (have a zero mean and one unit variance). Secondly, it is assumed that the features are measured without error and are reliable.

4.1 Normalisation

The range of values determined by the above mentioned features varies widely. The regression models that will be considered, all prefer the data to be normalised. If the data is not normalised, it could result in distorted relationships between the features and the value variable [18]. Feature normalisation involves manipulating the feature set to have a zero mean and variance of one.

The goal of standardizing the feature set, is to ensure that the features are in similar ranges. Additionally, standardizing the data allows algorithms such as gradient descent (used in linear regression) to converge faster, and leads to improved performance in algorithms such as Support Vector Regression [19].

4.2 Feature Selection

Reducing the dimensionality of the feature space, results in faster performance for the regression and classification models, as well as a lower variance in the data which means the models can better generalise [20].

A linear regression test is applied to each feature, to establish its F-score [21]. The test works by orthogonalizing the regressor and the data, then computing the correlation between the regressors and finally calculating the F-score. The top K (six in our case) of the features are then selected and are cross multiplied to form $K!$ new features which are then added to the existing data. When the algorithm was run on the quality-based features, it identifies the following six best predicting features: readability, sentiment, subjectivity, thread relevance, timeliness and engagement.

5. REGRESSION MODELS

We apply various regression techniques to determine the predicted community rating of an unrated comment. Regression is a statistical processes to estimate a relationship between variables. In this case, a regression model will help estimate the relationship between a set of features and a score. After a relationship has been estimated, the model can be used to predict the value when presented with a new feature set.

Two regression models will be compared to determine which model best fits the data domain and performs the best, as well as which model gives the highest prediction accuracy. The regression models that will be considered are:

- linear regression [22] and
- support vector regression (with rbf kernel) [23]

The alternative to the regression approach, would be to discretise the continuous value of the regression variable into classes, and using it as input for classification algorithms. For both approaches to determine the dependent variable, the continuous value was binned into sets of two , three and five classes respectively.

[2] This was chosen as a training set, as it is the closest training set we could find that relates to comments.

As with regression, the classification algorithms are instances of supervised learning techniques that trains on a specified training set of features and classification variables.

Four classification algorithms will be compared and experimented with and evaluated accordingly. The classifiers that will be considered are:

- support vector classification (with rbf kernel) [24],
- support vector classification (with linear kernel),
- logistic regression [25], and
- random forest classification [26].

Both the regression and classification models were implemented with the Scikit-Learn Python library [27].

6. EXPERIMENTS

For all the experiments discussed below, regression, as well as classification with two, three and five balanced classes, are compared. The regression models are scored by doing a 50 fold cross-validation and taking the mean R^2 score [28][3] of all the folds. The classification experiments are evaluated with an accuracy score[4].

We investigate another measure of predictive accuracy by ranking a list of comments using some ordering. This ranking is then compared to an ideal ordering, determined by ranking the same set of comments by community ratings (like to dislike ratio).

The correlation of the two rankings are measured using normalized discounted cumulative gain (NDCG [29]). NDCG reflects the intuition that accuracy at the top of the list is more important than ranking errors further down the list, which fits the comment ranking model well [2]. NDCG gives a score ranging from 0 to 1, where a higher score indicates a greater correlation between the predicted rank order and the ideal rank order.

6.1 Experimental Setup

For regression, we compare three different types of feature sets and both the vote ratio and percentage of votes are used as the dependent variables. For the classification experiments, only the vote ratio is investigated as a dependent variable.

Firstly, the quality-based feature set, mentioned in Section 3, is used, as well as the features obtained through feature selection (the top six features explain 70% of the variance and are cross multiplied to form extra features). Only comment threads with more than 50 comments are considered. Individual comments are disregarded for the training set, if they have less than five likes or dislikes respectively, less than 50 combined likes and dislikes, or contain less than 100 words. This results in a feature set containing 10296 objects, each consisting of 40 features. The training and test sets make up 67% and 33% of the feature set, respectively.

Secondly, the content-based feature set (a bag-of-words vector) is used. The vector consists of the 100 most used words in the comment space and the frequency at which each comment uses those words.

The final feature set consists of the bag-of-words vector, concatenated to the extracted feature vector.

6.2 Results

Feature Set	% of Votes	Vote Ratio
Quality-Based	**0.152**	0.029
Content-Based	0.032	0.116
Quality + Content	0.150	0.125

Table 1: Linear Regression Result Summary Table.

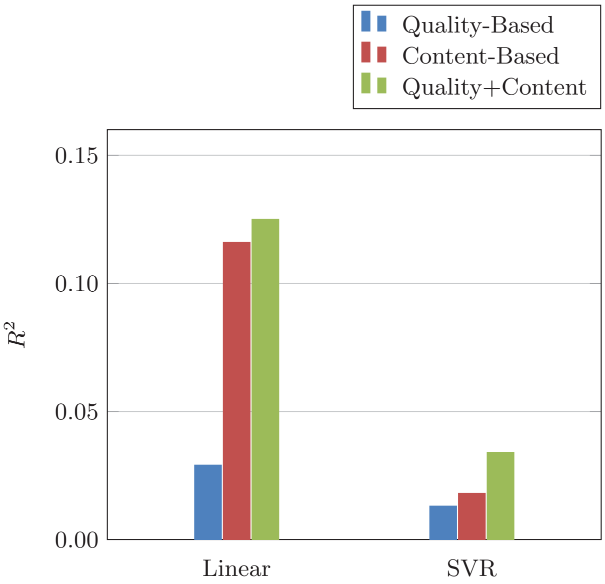


Figure 2: Regression results for training on like-to-dislike ratio.

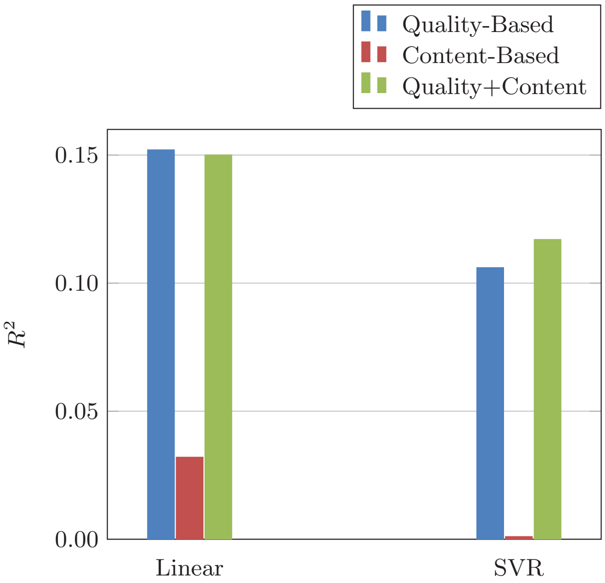


Figure 3: Regression results for training on percentage of total number of votes.

[3] a value between zero and one where a higher value represents better predictive accuracy

[4] The percentage of samples correctly classified. Thus, a higher value represents better predictive accuracy

Figure 2 shows that quality-based features are insufficient to predict community preference (when using vote ratio), but that content-based features, as well as quality- and content-based features combined, show better performance.

Figure 3 shows that quality-based features are better suited for predicting engagement (percentage of votes). Augmenting the quality-based features with the content-based features yields similar results.

Algorithm	Two Types	Three Types	Five Types
SVC	0.594	0.425	0.253
Linear SVC	0.616	0.439	0.285
Random Forest	0.588	0.412	0.265
Logistic Regression	0.619	0.438	0.289

Table 2: Results for classification on the quality-based features.

Algorithm	Two Types	Three Types	Five Types
SVC	0.661	0.474	0.314
Linear SVC	0.652	0.466	0.314
Random Forest	0.637	0.430	0.278
Logistic Regression	0.653	0.468	0.309

Table 3: Results for classification on content-based features.

Algorithm	Two Types	Three Types	Five Types
SVC	0.647	0.454	0.313
Linear SVC	0.662	0.455	0.321
Random Forest	0.622	0.410	0.280
Logistic Regression	0.662	0.451	0.323

Table 4: Results for classification on both quality- and content-based features.

The results in Tables 2 to 4 show that the accuracy of the classifier degrades as the data is segregated into more categories, as is expected. It is also evident that, in general, Support Vector Classification performed better than the other models. Table 3 shows that SVC obtains an average accuracy of 47.4% with classifying on the bag-of-word vectors. This is almost as accurate as the classification scores obtained by Wanas et al. [9] (49%), and is deemed sufficiently accurate given the context.

The content-based feature classification clearly outperforms classification on quality-based features, but also when the quality-based features are added to the content-based features.

Training the regression model on total number of votes, rather than the like-to-dislike ratio, results in significantly higher R^2 scores in the regression experiments, indicating that the total number of votes is a better indicator of community preference.

Table 5 shows that the regression accuracy increases logarithmically with the content-based regression model, as the size of the vector increases. This shows that even better results are possible with larger vector sizes, but should plateau and diminish when the vectors become too sparse.

Vector Size	R^2
50	0.061
100	0.115
200	0.129
500	0.178
1000	0.181

Table 5: Linear Regression on word vectors of different sizes.

6.3 Rank Correlation

Using the trained classifiers, we impose an ordering (or ranking) on a set of comments. This ranking is then compared to an ideal ranking with the NDCG measure.

For the experiment, a linear regression classifier is trained with a training set consisting 19014 comments. The NDCG score is computed with a K value that indicates how many from the list is considered for the comparison (we used $K = 20$). The model is trained and tested with 20-fold cross validation, so the NDGC scores reported in Table 6 is the mean of 20 recorded scores. The classifier is used to predict and rank the list of comments, and the comments' real community like-to-dislike ratio (as in Section 3.3) is used as the ground-truth ordering. NDCG scores range from 0 to 1, where a higher NDCG score indicates that the list ordering in question correlates well to the ideal ordering (i.e. ordered by vote ratio). Table 6 shows that content-based features correlate better to the community ordering.

Quality-Based	Content-Based	Both
0.597	0.782	0.759

Table 6: Normalized Discounted Cumulative Gain with different feature sets for the classifier predicted comments against the community ranked comments.

Further, Figure 4 shows how other orderings compare to our classifier orderings. The 'Random Ordering' simply imposes a shuffle on the comments and runs the NDCG algorithm on the result (intuitively, this should give a lower NDCG score, since the order is arbitrary). The 'Timestamp Ordering' ranks comments in the order that they arrived on the website, with the oldest comment being ranked first (similarly, the order is arbitrary regarding comment popularity, so it should give a lower score).

Figure 4 shows that ordering the comments according to date, or randomly, results in a list that does not correlate well with the community preference, for any feature set. What is encouraging, is that our proposed automatic ranking algorithm performs much better than the other two orderings when the bag-of-words feature vector is used, and according to Table 6, shows comparable performance to the classifier designed by Hsu et al [2].

7. PREVIOUS WORK

Our work in this paper is based on previous studies of comment ranking techniques by Lampe and Resnick [30], Wanas et al. [9], and Hsu et al. [2].

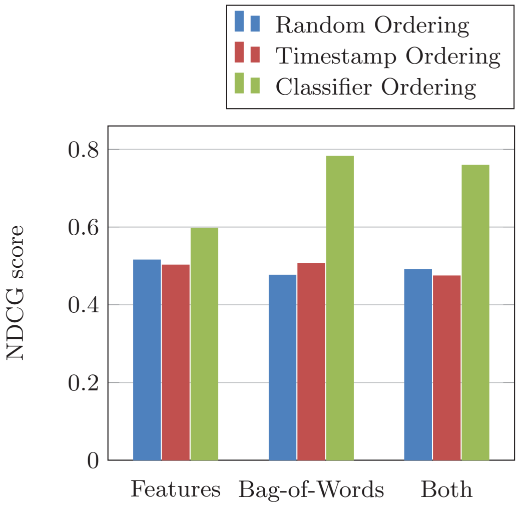


Figure 4: A comparison of NDCG scores for the different feature sets and different list ordering schemes.

7.1 Community Moderation

Lampe and Resnick [30] asked the question: "Can a system of distributed moderation quickly and consistently separate high and low quality comments in an online conversation?". Their analysis showed that a system that uses the participants in an online conversation as moderators, can efficiently rank comments so as to improve the quality of the conversation. They focused their investigation on slashdot.org.

Firstly, they used the properties of the comments left by users (comment length, word usage), as well as the properties of the authors themselves (frequency of posting, frequency of response) as ways to classify comment. They then found that the judgements of other users were better indicators of which comments needed attention.

Their investigation then involved building a regression model that predicted the final score of an unmoderated comment (what we based our models on), based on the classified comments that the users provided.

7.2 Automatic Scoring and Classification

Wanas et al. [9] seeked to improve on the work done by Lampe and Resnick [30]. The latter's rating system noticed that a significant amount of time had to pass before users could identify good quality comments. Additionally, earlier posts received more attention. Wanas et al. proposed a scheme of automatic post ranking based on supervised learning techniques (Support Vector Classification). Similar work was done by Hsu et al. [2], but using Support Vector Regression.

The features that Wanas et al. used, were based on features designed by Weimer et al. [10], and consisted of various features categorised into five classes. Those classes were relevance, originality, forum-specific, surface (frequency of capitalised words, quality of grammar, etc.) and posting component features (presence and quality of weblinks in posts). The trained classifier designed by Weimer et al. merely classified posts as 'bad' or 'good' and required that posts used as training data observe proper use of language and linguistic rules. As observed by Wanas et al, this is not always the case in online forums. They focused their investigation on providing finer ratings for posts, as well as taking various linguistic phenomena that frequent online forums, into account.

Their experiments showed their classifier to be 49.5% accurate when classifying posts as bad, average and good (in terms of their definition of quality). They claim the accuracy to be sufficient to provide rankings for posts. Their experiments also showed that structural features of posts were more significant in classification than features analysing the actual text. This means language independent approaches could be adopted, and led us to investigate improving upon quality-based features with content-based features.

8. CONCLUSION AND FUTURE WORK

The regression and the classification results show that the quality-based features lack in predicting the community popularity of a comment. This could be attributed to biased voting patterns in the community, eg. users that would "like" a comment multiple times if it supports their viewpoint (politically, religiously, or otherwise), but not necessarily evaluate the comment's quality. Using content-based features performs significantly better and allows us to achieve high comment rank correlation (NDCG) to the community's preference.

The quality-based features are, however, better suited for predicting the engagement a comment will receive from users in a comment thread.

Future expansions of this research will include designing specific features for the language domain, that incorporate a list of profanities specific to South African English.

The investigated models will also be trained and tested on comments scored by independent editors. We predict that the quality-based features should perform better when predicting editor preference, since it would represent the perceived ordering of comments according to the designers of the commenting system and their desire for what the quality of the comments should be.

9. ACKNOWLEDGEMENTS

Some of this work was done at the MIH Media Lab at Stellenbosch University.

10. REFERENCES

[1] N. Diakopoulos and M. Naaman, "Towards quality discourse in online news comments," in *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pp. 133–142, ACM, 2011.

[2] C.-F. Hsu, E. Khabiri, and J. Caverlee, "Ranking comments on the social web," in *Computational Science and Engineering, 2009. CSE'09*, vol. 4, pp. 90–97, IEEE, 2009.

[3] S. Keibler, "Importance of the online news portal." `http://www.buddy4study.com/blog/importance-online-news-portal`. Accessed: March 2014.

[4] M. P. OâĂŹMahony and B. Smyth, "A classification-based review recommender," *Knowledge-Based Systems*, vol. 23, no. 4, pp. 323–329, 2010.

[5] M. Rowe, S. Angeletou, and H. Alani, “Predicting discussions on the social semantic web,” in *The Semanic Web: Research and Applications*, pp. 405–420, Springer, 2011.

[6] D. M. Lane, *Online Statistics Education: A Multimedia Course of Study*, ch. Introduction to Linear Regression. Rice University and National Science Foundation, 2004.

[7] P. University, *WordNet 3.0 Free Dictionary.* March 2014. Statistical Regression.

[8] M. Karagiannopoulos, D. Anyfantis, S. Kotsiantis, and P. Pintelas, “Feature selection for regression problems,” *Proceedings of HERCMAâĂŹ07*, 2007.

[9] N. Wanas, M. El-Saban, H. Ashour, and W. Ammar, “Automatic scoring of online discussion posts,” in *Proceedings of the 2nd ACM Workshop on information Credibility on the Web*, pp. 19–26, ACM, 2008.

[10] M. Weimer, I. Gurevych, and M. Mühlhäuser, “Automatically assessing the post quality in online discussions on software,” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 125–128, Association for Computational Linguistics, 2007.

[11] P. Norvig, “How to write a spelling corrector.” `http://norvig.com/spell-correct.html`. Accessed: March 2014.

[12] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python.* O’Reilly Media, Inc., 2009.

[13] A. U. Alvarez, “Bad words list.” `http://urbanoalvarez.es/blog/2008/04/04/bad-words-list/`. Accessed: March 2014.

[14] A. Aizawa, “An information-theoretic perspective of tf–idf measures,” *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.

[15] R. Flesch, “A new readability yardstick.,” *Journal of applied psychology*, vol. 32, no. 3, p. 221, 1948.

[16] Ranks.nl, “Default english stopwords list.” `http://www.ranks.nl/stopwords`. Accessed: July 2014.

[17] M. F. Porter, “An algorithm for suffix stripping,” *Program: electronic library and information systems*, vol. 14, no. 3, pp. 130–137, 1980.

[18] J. Osborne and E. Waters, “Four assumptions of multiple regression that researchers should always test,” *Practical Assessment, Research & Evaluation*, vol. 8, no. 2, pp. 1–9, 2002.

[19] R. Herbrich and T. Graepel, “A pac-bayesian margin bound for linear classifiers,” *Information Theory, IEEE Transactions on*, vol. 48, no. 12, pp. 3140–3150, 2002.

[20] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[21] S. MacKenzie, “How to report an F-statistic.” `http://www.yorku.ca/mack/RN-HowToReportAnFStatistic.html`. Accessed: April 2014.

[22] B. Flury and H. Riedwyl, “Multiple linear regression,” in *Multivariate Statistics*, pp. 54–74, Springer, 1988.

[23] D. Basak, S. Pal, and D. C. Patranabis, “Support vector regression,” *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.

[24] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features.* Springer, 1998.

[25] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Introduction to the logistic regression model.* Wiley Online Library, 2000.

[26] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, “Random forest: a classification and regression tool for compound classification and qsar modeling,” *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[28] A. Colin Cameron and F. A. Windmeijer, “An r-squared measure of goodness of fit for some common nonlinear regression models,” *Journal of Econometrics*, vol. 77, no. 2, pp. 329–342, 1997.

[29] W. B. Croft, D. Metzler, and T. Strohman, *Search engines: Information retrieval in practice.* Addison-Wesley Reading, 2010.

[30] C. Lampe and P. Resnick, “Slash (dot) and burn: distributed moderation in a large online conversation space,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 543–550, ACM, 2004.