Comparison of Quality Indicators in User-generated Content Using Social Media and Scholarly Text

Renhao Cui
The Ohio State University
renhao.cui88@gmail.com

Manirupa Das
The Ohio State University
manirupa@gmail.com

Abstract
Predicting the Quality of a text document is a useful task when facing the problem of measuring the performance of this document before release. In this project, we test on different features including textual and Meta ones, and show that the performance of the prediction relies differently on these features. Moreover, we also compare the result on both structured data such as academic articles and unstructured data such as tweets, and show that the same method could perform differently across domains.

1 Introduction

For the past years, the amount of open domain data has dramatically increased. Among all the data, textual data is still one of the most important and informative ones. While social media is getting more popular these days, standard text content is still a critical type of method to provide information and ideas. Structured data such as news and academic articles has been studied well in terms of summarization, topic modeling, opinion mining, information extraction, etc. On the other hand, unstructured data such as tweets, statuses, and text messages is adding more challenges to the classical task, especially when considering the out-of-vocabulary words, the length, and nonstandard grammar. Among all the tasks, predicting the quality of a text document is an interesting and useful proposition. In this project, we would like to explore the different factors that affect the quality of a document, as well as a comparison between structured data and unstructured data towards the role of each factor. The quality of a text document is a very subjective measurement, so in this project we use the Pagerank score as class label for scholarly articles dataset and number of retweets as the indicator of quality for the tweets dataset.

2 Methodologies

We cast the problem of predicting a document's quality or importance as a task in supervised learning and make use of several methods that we apply towards feature value calculations and towards training of our datasets.

2.1 Topic Modeling

Topic modeling assumes that each word carries some meaning towards a set of topics, then each document, treated as a combination of the words, carries some meaning related to the set of topics as well. It is represented as a probability distribution across the set of topics given each document. We use Latent Dirichlet Allocation [1] to model the documents and find the potential topic distribution.

2.2 Part-of-speech tagging

Part-of-speech tagging is used to distinguish the role of the word within a sentence, and to better understand the structure of the sentence by annotating each word with a part-of-speech marker. Some words have multiple part-of-speech tags, so context is brought into the consideration for figuring out its POS tag. In general, a chain model that carries the information from the previous predictions in marking the current one is used to consider the context information.

2.3 Readability Score

Readability scores measure how easy it is to read and understand a document. It tries to predict the reaction of humans as they read the document. In this project, we consider several metrics. The Coleman–Liau Index of readability measurement [2] is calculated as: **CLI = 0.0588L - 0.296S - 15.9** where L is the average number of letters per 100 words and S is the average number of sentences per 100 words.

The ARI metric [3] where the authors analyzed samples of the textbooks used in the Cincinnati Public School System from which they derived the multiple regression formula given by: **ARI = 4.71 (characters/words) + 0.5**

(words/sentence) − 21.43, uses the average number of words per sentence and the average number of characters per word to estimate the readability.

The GFI metric [3] given by **GFI = 0.4 (words/sentences + 100 complexwords/words)**, uses the average number of words per sentence and the average number of complex words in the text, such that short sentences in plain English achieve a better score than long sentences written in a complicated language. A complex word is defined as a word with three or more syllables.

2.4 Sentiment Analysis

Sentiment analysis is used to extract subject information from text content, and determine the attitude of the author with respect to the overall polarity of the text. Generally this type of analysis is performed on a word-to-word or sentence basis, and then combined together as the sentiment of a document. Various models have been applied to this problem, and most of them are treated as a classification problem, with a deterministic or probabilistic output indicating the polarity.

2.5 Logistic Regression

In recent years there has been extensive use of conditional or discriminative probabilistic models in NLP, IR, and Speech, because they give high accuracy performance, they make it easy to incorporate lots of linguistically important features and they allow automatic building of language independent, retargetable NLP modules [4]. Discriminative (conditional) models take the data as given, and put a probability over hidden structure given the data, In this project, we use logistic regression as the predictive model for our experiments as it is useful in estimating the parameters of a qualitative response model [5].

The model can be used for predicting the outcome of a categorical dependent variable, based on one or more explanatory variables (features). The probabilities describing the possible outcomes of a single trial are modeled as a function of the explanatory (predictor) variables or features, using a logistic function [5]. The logistic function is useful because it can take an input with any value from negative infinity to positive infinity for the features, whereas the output F(t) always takes on values between 0 and 1 [5].

$$F(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}},$$

Thus the Logistic function squashes linear response into a value between 0 and 1, and hence is interpretable as a probability [5]; therefore we label the output as the class with the higher probability.

3 Dataset

3.1 Data Preparation

3.1.1 Tweets

We extract English tweets through Twitter streaming API across 7 days, and result in about 3.5 million tweets. Since the task is to predict the influence of a tweet, so we only pick the original tweets, so we filter out duplicates and strange-formatted tweets that are not easy to process. We use the number of retweets as the class label, and format a binary decision per previous work: whether this tweet got any retweets (class 2) or not (class 1) [6]. Finally we trim the dataset to balance the data for both labels, and it results in 110k tweets for this project.

3.2.2. Articles

The ACL Anthology [7] is a digital archive of conference and journal papers in natural language processing and computational linguistics whose primary purpose is to serve as a reference repository of research results, but the creators believe that it can also be an object of study and a platform for research in its own right and have made this dataset available as a freely downloadable resource that can be used for research in scholarly document processing. We use the latest 2013 release of this dataset, spanning 21,212 papers, 17,792 authors, 342 venues and 110,975 citations.

The Pagerank statistic of the article was used as the class label for the articles since it measures the importance of an article within a network. Here we looked at the real values of this statistic and choose the threshold as the lowest *positive* value of Pagerank that corresponds to 4163 articles out of the 11592 articles, to convert this to a categorical value giving a binary decision: well-cited paper (Y) versus not well-cited paper (A). We trim the dataset to balance both the labels, and since only papers with positive values of Pagerank were chosen every paper considered has definitely been cited, hence the binary decision is for papers of moderate importance (not well-cited A) versus high importance (well-cited Y), and this results in a dataset of 11592 articles for this project.

3.2 Feature Extraction

3.2.1 Meta Feature

Meta feature is used to describe some high level characteristics other than textual content. In this project, for tweets, we use the number of followers, the number of favorites, and the number of posted tweets of the author as Meta features. These can represent the popularity of the author, which related to how many people can see the posts, and the influence of the author. Since we use the number of retweets as the label, these features can also been used to balance the activity level of the author. For the scholarly article dataset, the meta feature set includes fields such as author identifier, author H-Index value, publication year, and publication venue, and it also includes some network-based features such as author in-citations, author out-citations, author Pagerank, paper betweenness centrality, paper closeness centrality, paper degree centrality, paper in-citations and paper out-citations.

3.2.2 Textual Feature

Text features are those extracted from the textual content of the articles or tweets [3]. We extract two kinds of textual features from the documents, a document coherence feature and a part-of-speech feature. Apart from this we also include length features such as word counts, character counts and sentence counts to represent the text.

3.2.2.1 Coherence/Topic Feature

In general, a document focuses on several key ideas that are used to express the opinion of the author. We believe the way the author organize these ideas is related to the quality of the document, so it will affect the influence of the document. To capture the arrangement of these ideas, we use topic modeling to generate the distribution of topics for a document. In this case, we treat this probability distribution as the representative of the arrangement of the ideas. Since most documents focus on a small number of topics, and to prevent noisy information as well as help the performance, we decide to take the normalized top-5 topic distribution as the coherence feature by default, although we perform the topic modeling for 30 topics. For this project, we use Stanford Topic Modeling Toolbox [8] to perform the LDA model training. Thus we have two sets of textual “coherence” features, the set of 30 topic distributions, and the set of top-5 normalized topic distributions for each document.

3.2.2.2 POS Feature

POS tags are used to capture the role and type of the words within the sentence. However, structured or formal and unstructured or informal documents have very different writing style, so we decide to use domain specific POS tags for this task. For articles, we use the standard Penn Treebank POS tag set available within the NLTK package [9] used for generating POS statistics for the article corpus with a total of 38 tags. On the other hand, we use twitter-based POS tagger from O’Connor to generate the tags for tweets, with a total of 24 tags [10].

3.2.2.3 Length Feature

Document length varies from one to the other, and it is closely related to whether the reader would prefer the content. It is also critical to the POS feature above, as we do not normalize the feature, thus we put the length as one of the features instead. This includes the sentence counts, word counts and character counts per document in case of articles, and the word token counts in case of tweets.

3.2.3 Readability Feature

It is not necessarily true that an easily understandable document has more influence among the readers. However, we think there is some relation between these two, so we use Coleman–Liau index as the readability feature. For the scholarly articles we also calculate and include the Automated Readability Index and Gunning-Fog Index readability measures described earlier as we want to take the role of average words/sentence, average characters/word and complex words into account for longer structured text such as articles.

3.2.4 Sentiment Feature

Sentiment analysis is used to analyze the opinion that the text expresses, which is positive or negative. To be more specific, a binary indication of the sentiment is too general for this task, so we use a confidence/probability score to represent the sentiment of the text. Therefore it gives more quantitative measurement of the sentiment of the document. For tweet dataset, we use a sentiment wordlist with probabilities created using a Language-independent Bayesian method from Davies and Ghahramani [15]. In the case of articles, we used the TextBlob API [16] for processing of the text for extracting the sentiment feature, and we consider two measures of sentiment, the average sentiment over all sentences in the article and also the maximum sentiment or maximum value of polarity for that article.

4 Experiments

4.1 Models

Based on the individual features mentioned in the previous section, we separate them into 4 categories:

1. ***Metadata/Network Feature*** - features that are not related to the actual text content of the document, and used to describe some background of the author and the document (tweet or article) and their peer network. In this project, it includes the Meta feature.
2. ***Textual Feature*** - features that are directly related to the text content of the document. In this project, it includes Coherence/Topic Feature, POS feature, and Length feature.
3. ***Readability Feature*** - this feature represents the degree to which a reader can read and understand the document. In this project, it includes the readability feature.
4. ***Sentiment Feature*** - this feature presents the sentimental opinion of the document. In this project, it includes the sentiment feature.

In order to explore the role of different features as well as categories, we run the experiments on the following 16 models.

Model No.	Description	Model No.	Description
1	Meta/Network Feature	9	Full Model without Coherence/Topic Feature
2	Readability Feature	10	Full Model without POS feature
3	Sentiment Feature	11	Full Model without Length Feature
4	Textual Feature (Coherence/Topic Feature + POS Features + Length Feature)	12	Full Model without Meta Feature
5	Full Model (Meta Feature + Readability Feature + Sentiment Feature + Textual Feature)	13	Coherence/Topic Feature with 5 topics
6	Full Model with 30 topics in Coherence/Topic Feature	14	Coherence/Topic Feature with 30 topics
7	Full Model without Readability Feature	15	Textual Feature + Sentiment Feature
8	Full Model without Sentiment Feature	16	Meta Feature + Sentiment Feature

4.2 Machine Learning Toolkit

We used the Weka data mining toolkit [11] - a collection of machine learning algorithms for data mining tasks, to run the logistic regression classifier. We use the default settings for the algorithm, for maximum iterations=1 and a ridge log likelihood value of 1.0E-8 and used 10-fold cross validation for our training and testing dataset split. We record results for metrics such as Precision, Recall, F-score for each class and the weighted average for each, and also the overall accuracy of the model.

5 Results

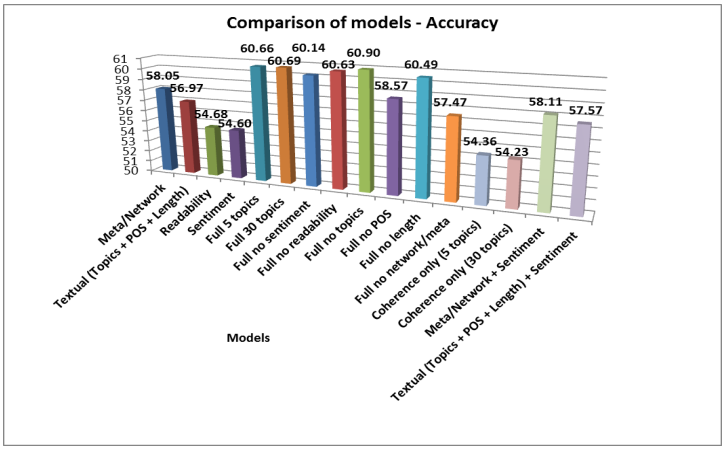
5.1 Tweets dataset

We run the models on the 110k dataset, and record the performance of accuracy, F1 scores for class1 (no re-tweet) and class 2 (got re-tweeted), average F1 score, average precision and recall. The data is all labeled with 54.6% in class 1.

5.1.1 Overall Performance

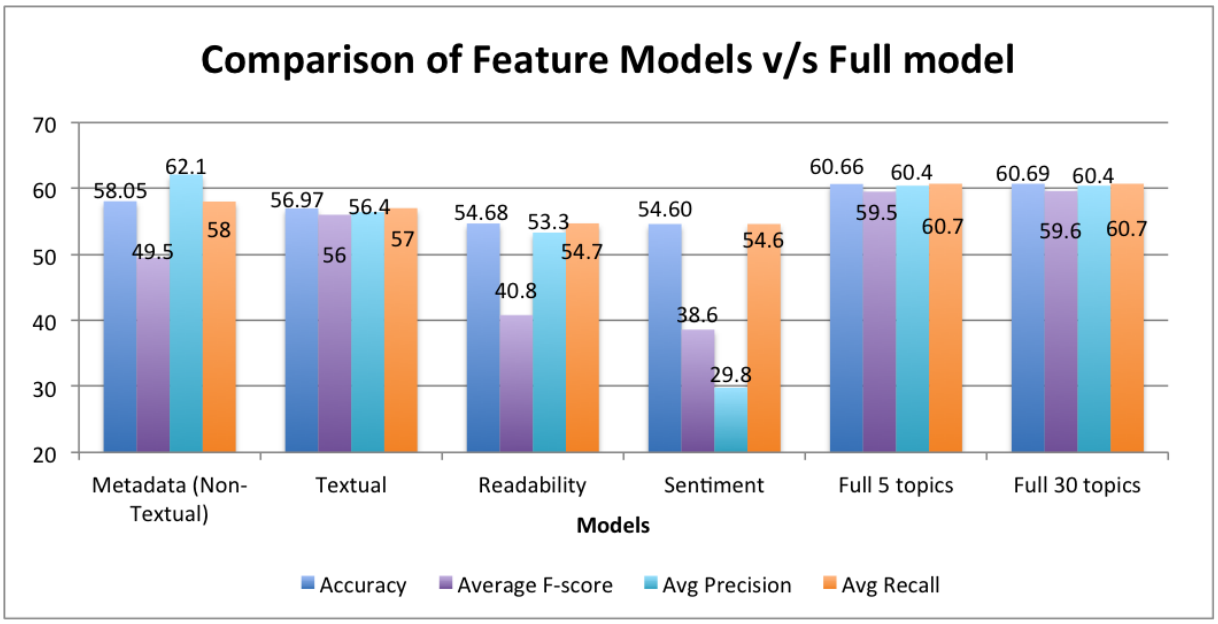
The full results on the 16 models are as follows:

Features	Accuracy	F-score Class 1	F-score Class 2	Avg F-score	Avg Precision	Avg Recall
Meta/Network	58.0476	71.1	23.5	49.5	62.1	58
Textual (Topics + POS + Length)	56.9704	64.2	46.1	56	56.4	57
Readability	54.6767	70.2	5.4	40.8	53.3	54.7
Sentiment	54.5963	70.6	0	38.6	29.8	54.6
Full 5 topics	60.6557	67.7	49.7	59.5	60.4	60.7
Full 30 topics	60.6928	67.6	50	59.6	60.4	60.7
Full no sentiment	60.1415	67.5	48.5	58.9	59.9	60.1
Full no readability	60.6259	67.7	49.6	59.5	60.4	60.6
Full no topics	60.8979	68	49.7	59.7	60.7	60.9
Full no POS	58.5681	69.5	35.5	54.1	59.2	58.6
Full no length	60.4949	67.6	49.5	59.4	60.2	60.5
Full no network/meta	57.4656	64.3	47.5	56.6	56.9	57.5
Coherence only (5 topics)	54.3577	69.9	6.1	40.9	50.8	54.4
Coherence only (30 topics)	54.2348	69.5	8.8	41.9	50.8	54.2
Meta/Network + Sentiment	58.1091	71.1	23.9	49.7	62.1	58.1
Textual (Topics + POS + Length) + Sentiment	57.5686	64.4	47.4	56.7	57	57.6



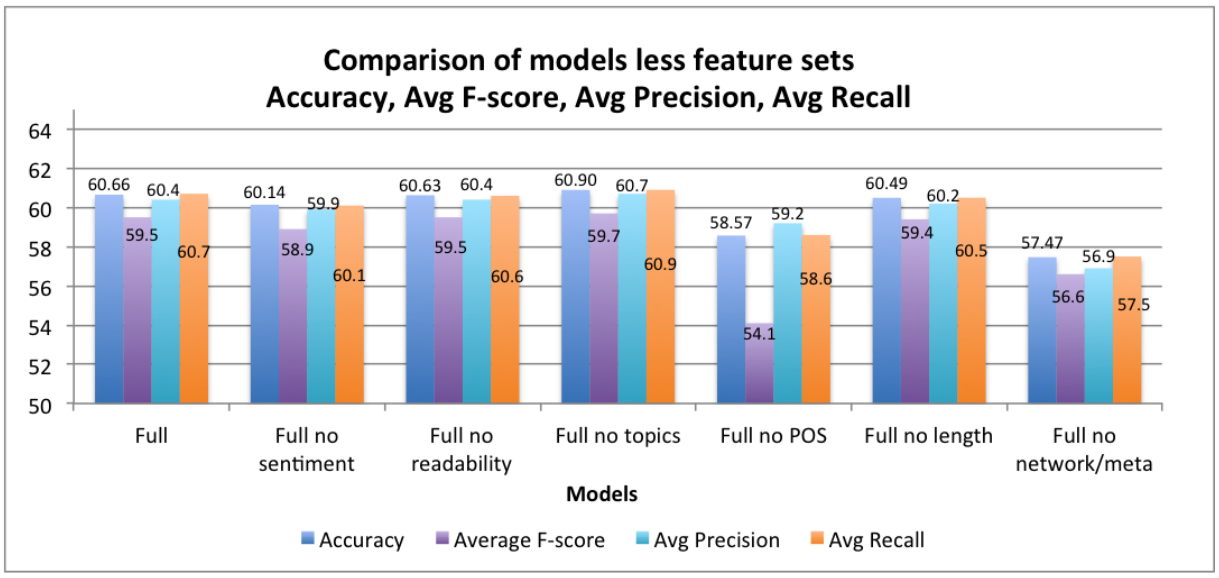
It can be seen that full model with 5 topics as the coherence/topic feature gives a relative good performance. However, some models outperform the full model. In other words, features play different roles in terms of the model performance. On the other hand, class 1 has a much better F-score compared with class 2. Class 1 is the one that has no retweets, so that it shows the system tends to predict a tweet to have no retweets.

5.1.2 Individual Feature Category



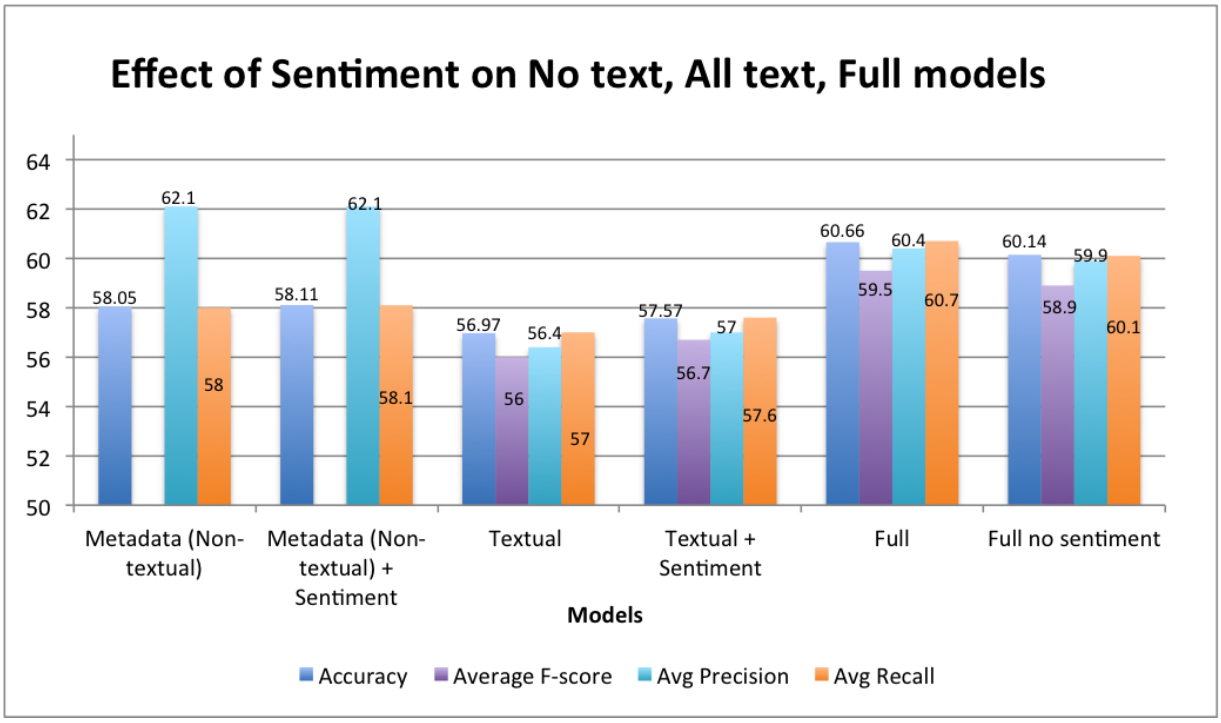
Overall, no single set of features outperforms the full model. But Meta feature has the best result among others, and it even gives a better result than the textual feature, which indicates that not knowing anything about the content but the author can still have a good prediction on the quality of the tweet. This is actually similar to some previous work that claims Meta or follower, retweet counts have a huge impact on the influence of the tweet [12][13]. Moreover, sentiment or readability feature only does not carry too much meaning, so that it cannot be used as a good single predictor as well. It also shows that with more topics as the summary of the content, it does not have a clear influence to the prediction, and the minor increase with more topics is not conceivable in this experiment.

5.1.3 Effect of Individual Feature



Starting from the full model, removing POS or Meta feature results in the largest decrease in performance. It shows that the Meta feature has a huge impact on the overall performance of the prediction. Meanwhile, POS feature, related to the textual content, also has a relatively big impact on the prediction, which shows that the use of proper type of words and maintaining a good structure of the sentence may gain some help in the quality of the tweet. On the other hand, removing readability and length features do not have much impact on the system performance, indicating these features are not critical in predicting the quality of tweets. Furthermore, removing the sentiment feature results in minor decrease, which agrees with the conclusion from the previous discussion that sentiment, does not give too much help with the prediction. However, removing topics feature actually increase the performance, which means that using 5 topics to summarize the text content does not help in predicting the quality. But the difference is not noticeable, so it cannot be considered as a hard conclusion.

5.1.4 Sentiment Feature



The comparison above strengthens the argument from the previous discussions that sentiment feature affect very little on the overall performance as a quality indicator. It can also be found that it has more positive impact on a model that has relatively poor performance (Metadata vs. Textual), while it has a negative impact on the full model which has the best performance among these three.

5.1.5 Summary

From the experiments above, we can see that using all features gives a relatively good performance. However, metadata only also has a huge impact on the prediction. Sentiment feature is clearly not a good indicator for this task, as well as readability and length feature since they do not have much influence either. Moreover, the use of topics as a summary does not play a good role as expected, and even has a negative impact in some cases, while the choice of number of topics does not have a clear difference as well.

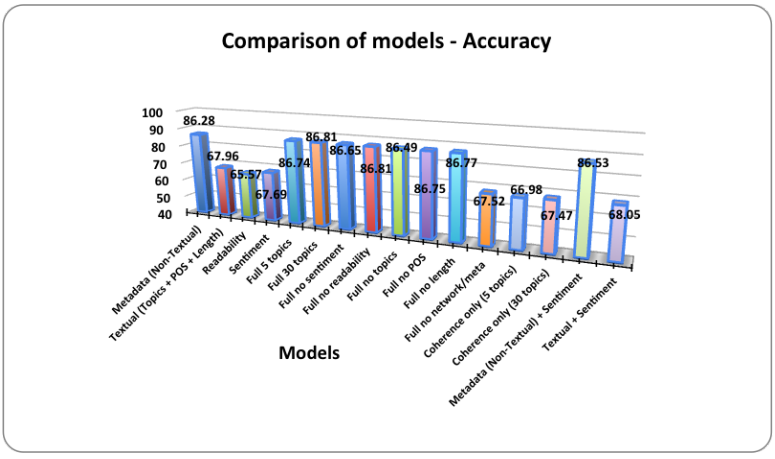
5.2 Articles dataset

We run the models on the dataset of 11592 articles with balanced Pagerank score, and record the performance metrics such as accuracy, F1 scores for each class, average F1 score, average precision and recall. In balancing the dataset our default or majority class turns out to be class Y, i.e. for the well-cited papers, with 64.08% of the dataset residing in this class. We perform most of our comparisons against this baseline.

5.2.1 Overall Performance

The results on the runs for the full set of 16 models are as shown below:

Features	Accuracy	F-score Class A	F-score Class Y	Avg F-score	Avg Precision	Avg Recall
Meta/Network	86.28	81.4	89.1	86.3	86.5	86.3
Textual (Topics + POS + Length)	67.96	26	79.6	60.3	70.5	68
Readability	**65.57**	27.9	**77.4**	59.6	63	65.6
Sentiment	67.69	25.2	79.4	59.9	69.9	67.7
Full 5 topics	**86.74**	81.8	89.6	86.8	86.8	86.7
Full 30 topics	**86.81**	**81.9**	**89.6**	**86.9**	**86.8**	**86.9**
Full no sentiment	86.65	81.7	89.5	86.7	86.8	86.6
Full no readability	**86.81**	81.9	**89.6**	86.9	86.9	86.8
Full no topics	86.49	81.5	89.4	86.5	86.6	86.5
Full no POS	**86.75**	82	**89.5**	86.8	87	86.7
Full no length	**86.77**	81.9	**89.6**	86.8	86.9	86.8
Full no network/meta	67.52	27.6	79.1	60.6	68	67.5
Coherence only (5 topics)	66.98	25.5	78.8	59.7	67	67
Coherence only (30 topics)	67.47	34.9	78.3	62.7	66.1	67.5
Meta/Network + Sentiment	**86.53**	81.8	89.3	86.7	**86.8**	**86.5**
Textual (Topics + POS + Length) + Sentiment	**68.05**	26.4	79.6	60.5	**70.6**	**68**



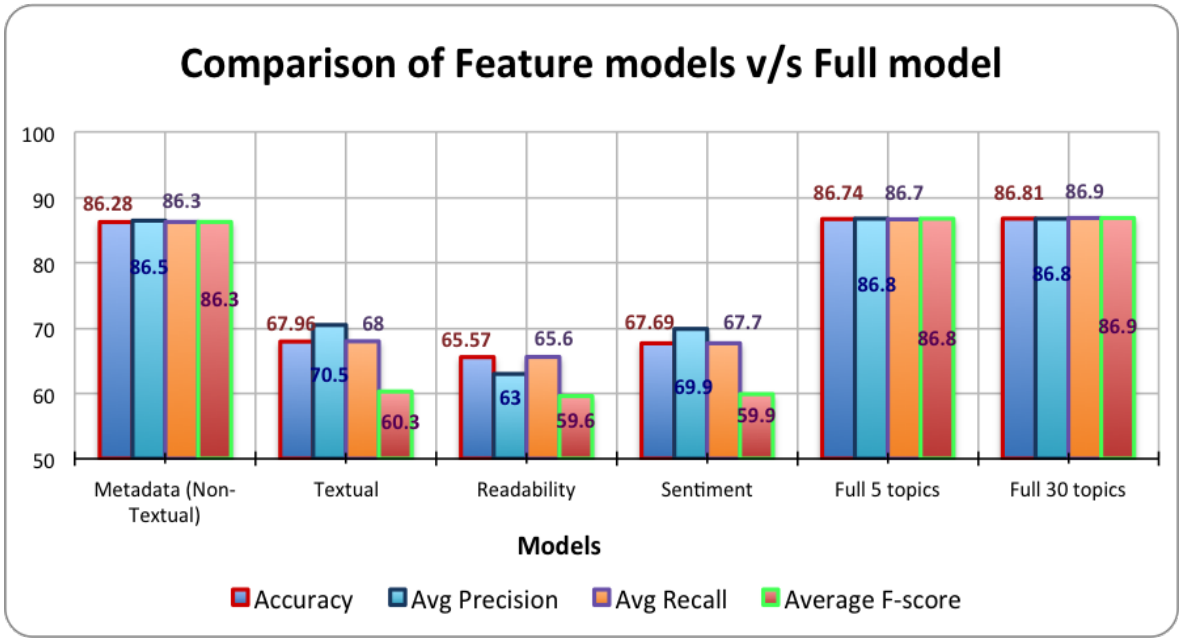
In the overall comparison we find that the Full model with 30 topics as the coherence feature gives the best overall performance at 86.81%. As in the case of articles, the majority class Y of well-cited papers has a much better F-score compared with class A of moderate importance papers. We know that ACL conference papers with Pagerank beyond our minimum positive threshold are likely to be well-cited within the community, i.e. at least better cited than the least cited papers within the community (i.e. class A), which our system correctly tends to predict.

In comparing with the individual feature models, no single model is able to surpass the performance of the full models with both 5 and 30 coherence features, though Metadata comes closest, at 86.28%. Readability, Sentiment and Textual models perform the least and do even worse than the majority baseline, though F-score shows that they do predict the well-cited class correctly with relatively high confidence. Comparing just the two Coherence features we find that the model with 30 topics does better overall than that with top 5 normalized topics.

Looking at the features suppressed from the full model (here we use the full model with 5 topics for coherence, at 86.74%), complementary to findings for individual models, we find that removing the Textual-based features such as Readability, POS and Length actually increases the performance of the system to 86.81%, 86.75% and 86.77%

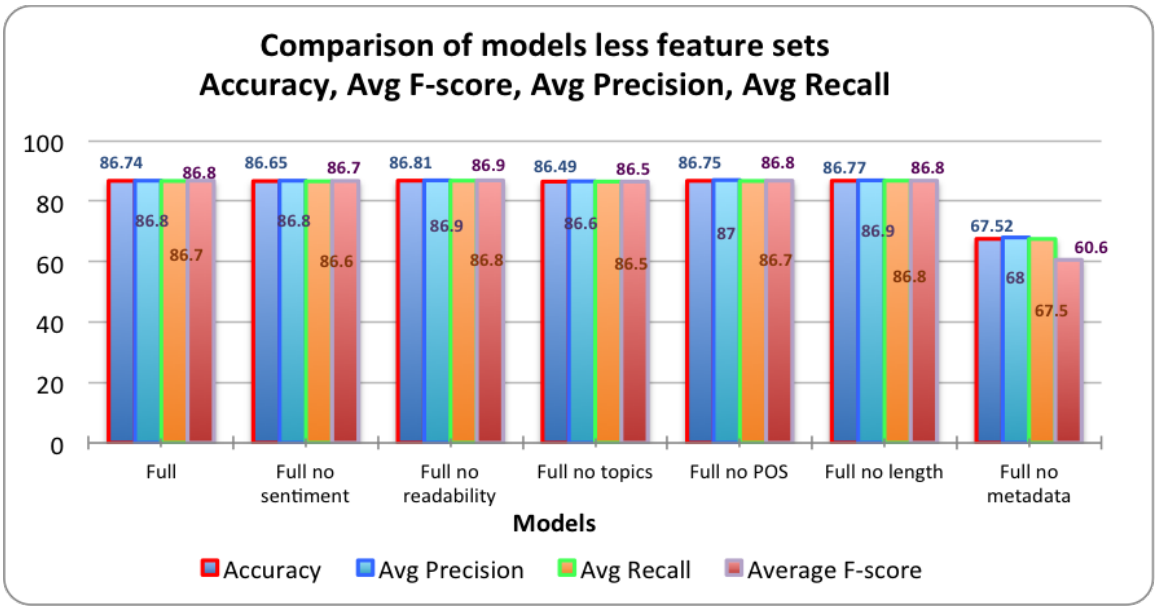
respectively showing that these are actually hurting performance, whereas taking away the coherence features reduce performance showing that these do contribute to the overall performance of the full model. As for the role of sentiment, we see that in each of the cases, Metadata, Textual and Full model, it contributes to the performance all-round, for each Accuracy, Avg. F-score, Avg. Precision and Avg. Recall.

5.2.2 Individual Feature Category



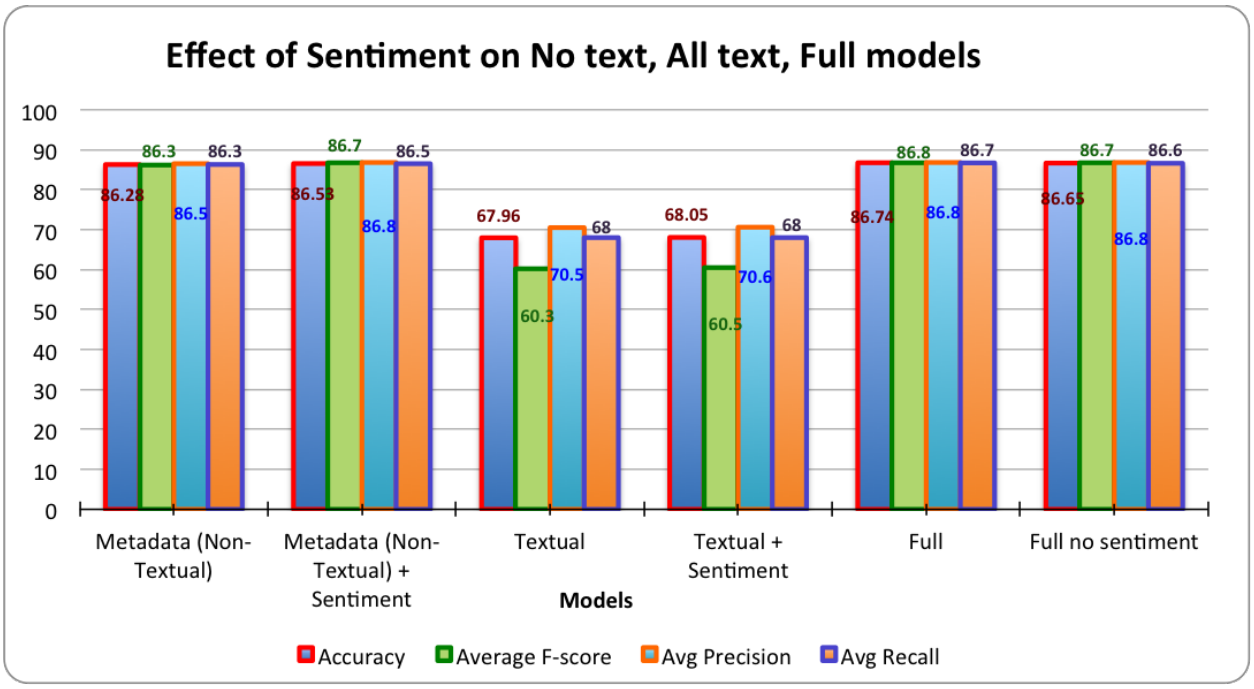
In this set of experiments we run the individual models for each feature set and compare the performance of each against the full model. Here we find that full model with the 30 topic features performs the best, with the non-textual metadata model performing the best among all the individual feature sets, with 86.28% accuracy. Sentiment and Readability models give the least performance, and do almost similarly with 67.69% and 65.57% accuracy respectively, and the Textual model does only slightly better than Sentiment and Readability with an accuracy of 67.96% and precision at 70.5% comparable to Sentiment at 69.9%.

5.2.3 Effect of Individual Feature



As seen previously, the removal of Metadata or Non-Textual features has the most negative impact on performance on the Full model, removing Coherence features (top-5 topics) also hurts performance, whereas removal of other textually based features like Readability, POS and Length actually increases performance indicating that these contribute little in terms of predictive value. The Full model also takes a hit on removal of sentiment, which hurts the performance.

5.2.4 Sentiment Feature



As we can see from the results and previous discussion, Sentiment adds predictive value on all counts to each of the models in consideration, Metadata (Non-textual), Textual (Coherence+POS+Length) and Full models. This highlights the fact that sentiment can play an important role in the how a scholarly article is received.

5.2.5 Summary

In the case of articles, Metadata and Coherence/Topic features contribute the most predictive value to the overall model, whereas other textually derived features such as Readability, Length and POS can hurt the performance. Contrary to the tweets domain, sentiment does contribute predictive value to the overall model for scholarly articles.

6 Conclusions

Besides the in-domain observation and analysis, there are also some interesting points that show difference or similarity across domains. In the project we find that for the tweet domain sentiment does not have a noticeable influence to the prediction of whether a post will be re-tweeted or not, however, for articles sentiment does bring an improvement over both Metadata (Non-Textual), Textual and Full models. This seems to follow intuition given that tweets are primarily an opinion-based domain where sentiment cannot add much more predictive value, whereas it adds a lot more predictive value to scholarly text. We also find that for both tweet and article domains Metadata or Non-Textual features play a significant role in boosting performance, followed by Coherence/Topic features, and features that related to the text content do not have the major effect as expected. This could be partially because people tend to relate the quality of a document with its author, or some other background factors. So in this project, this may be due to the use of class labels that bring out the bias towards non-textual elements of both datasets, hence a different label could be used to see if this offsets the role of Metadata and boosts the role of Textually-based features.

7 Future Work

One future direction is to identify a suitable baseline among all of these models to build on top of and come up with an optimal model that generalizes well to most datasets in the scholarly articles and tweets domains. Another direction is to make our comparison more effective by normalizing results across the domains in some way so as to highlight the differences better and make the comparison easier to interpret. For future extensions to this work we would like to include context features into our model e.g. referring texts such as new textual content added into a re-tweet for a new tweet and explore the role of citing sentences [14] for articles. Additionally we would like to look at methods to improve the predictive value of our textual features that are directly related to the content as quality indicators, by possibly deriving semantic features using WordNet [17] and other ontologies or using skip gram features [18]. Since our study is mainly about trying to understand which features from the text are most indicative of quality or importance of a document, we would also like to try our experiments by choosing different or more comprehensive labels than number of retweets and Pagerank, as there might be elements in the metadata that are biased towards predicting these quantities for both the domains. Thus, choosing the right elements, or the right combination of elements, to use as a gold label for our specific experiments, so as to help extract the desired level of signal from the textual features, could also be a course of future study.

8 Project Repository

Source and Data - http://tinyurl.com/n8uq8u9

References

[1] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *the Journal of machine Learning research* 3 (2003): 993-1022.

[2] Coleman, Meri, and T. L. Liau. "A computer readability formula designed for machine scoring." *Journal of Applied Psychology* 60.2 (1975): 283.

[3] Dalip, Daniel Hasan, et al. "Automatic assessment of document quality in web collaborative digital libraries." *Journal of Data and Information Quality (JDIQ)*2.3 (2011): 14.

[4] Klein, Dan, and Christopher Manning. "Maxent models, conditional estimation, and optimization." *HLT-NAACL 2003 Tutorial* (2003).

[5] "Logistic Regression." *Wikipedia*. Wikimedia Foundation, 12 Nov. 2014. Web. 11 Dec. 2014.

[6] Hong, Liangjie, Ovidiu Dan, and Brian D. Davison. "Predicting popular messages in twitter." *Proceedings of the 20th international conference companion on World wide web*. ACM, 2011.

[7] Bird, Steven, et al. "The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics." *LREC*. 2008.

[8] "Stanford Topic Modeling Toolbox." Stanford Topic Modeling Toolbox. N.p., n.d. Web. 12 Dec. 2014.

[9] "Natural Language Toolkit." Natural Language Toolkit — NLTK 3.0 Documentation. N.p., n.d. Web. 12 Dec. 2014.

[10] Chris, Olutobi Owoputi Brendan O'Connor, and Dyer Kevin Gimpel Nathan Schneider. "Part-of-speech tagging for Twitter: Word clusters and other advances." (2012).

[11]] Hall, Mark, et al. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter* 11.1 (2009): 10-18.

[12] Khabiri, Elham, Chiao-Fang Hsu, and James Caverlee. *"Analyzing and Predicting Community Preference of Socially Generated Metadata: A Case Study on Comments in the Digg Community." ICWSM*. 2009.

[13] Petrovic, Sasa, Miles Osborne, and Victor Lavrenko. "RT to Win! Predicting Message Propagation in Twitter." *ICWSM*. 2011.

[14] Abu-Jbara, Amjad, and Dragomir Radev. "Reference scope identification in citing sentences." Proceedings of ACL 2012: Human Language Technologies.2012.

[15] Davies, Alexander, and Zoubin Ghahramani. "Language-independent Bayesian sentiment mining of Twitter." (2011).

[16] "TextBlob: Simplified Text Processing." TextBlob: Simplified Text Processing — TextBlob 0.9.0 Documentation. N.p., n.d. Web. 12 Dec. 2014.

[17] Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. "WordNet:: Similarity: measuring the relatedness of concepts." *Demonstration Papers at HLT-NAACL 2004*. Association for Computational Linguistics, 2004.

[18] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in Neural Information Processing Systems*. 2013.