
Document Quality Prediction with Textual Features

Bing Bai

Division of Computer Science, 110 Frelinghuysen Rd., Piscataway, NJ 08854

bbai@paul.rutgers.edu

Abstract

This paper presents machine learning studies on how to predict pre-defined document qualities (not relevance) such as depth and objectivity with pure textual features. The data is a document set with 2200 documents whose qualities are judged by trained faculty members, professionals and students and textual features computed by GATE. Our work is to select an algorithm can give better predictions. The results of several algorithms (including ID3 decision tree, Naïve Bayes and a new algorithm we proposed) are presented.

1 Introduction

The usual concern in the document retrieval is the relevance of the text to the given topic. The research in the area has been used widely in applications such as the web search. With the increase of the size of the corpora, it becomes more and more important that we care about not only relevance, but also the “quality” of the documents. Qualities such as the depth and the accuracy of the document can also be indicators of the potential value to the topic, as well as the relevance.

Identifying the “quality” of the document seems to be a very hard problem. A good judgment may require an experienced professional with enough background information to carefully read the document, which is not feasible for huge corpus processing. However, human judgment is often traceable in an intuitive way. For example, some words like “coward” will usually lower the objectivity score in human judgment. This fact leads to an idea that it’s possible to predict some qualities using the textual features.

We used the following testing environment prepared by our colleagues (Tang 2003): 2200 documents are selected from both the TREC document set and the Internet, and 9 document qualities are defined. We only discuss 3 qualities in this paper: Depth, Multi-view and Objectivity. These qualities are judged for each document by college students and professionals. Each quality has a score on a 10-point Likert scale, in which 1 means lowest score and 10 means highest score, a special score “0” means “no basis for judgment”. After the “no basis for judgment

cases were removed, 2013 documents were left. The same experiment was done in both SUNY Albany and Rutgers, so I have two sets of scores for this 2013 document set. Meanwhile, 139 textual features are prepared with the natural language processing tool GATE (Cunningham, 2000) by Peng Song¹, Robert Rittman and other colleagues. The feature set included punctuation, acronyms, length of the document/title/subtitle, key words such as “say” or “seem”, etc. GATE simply counted the number of times these features appeared in the document.

Like many other machine learning problems, the difficulty is that there is considerable noise in the data. We use the average scores of Rutgers and Albany. Also, with an idea from Paul Kantor, we simplified this problem into a 2-value classification problem, if a score is higher than the average score of the corresponding quality, we set the “normalized score” to 1; and set the “normalized score” to 0 if the score is lower than the average.

The results of the classical machine learning methods showed fair performances. The typical predicted accuracy is from 55% to 68%. See Section 2 for details.

The distributions of the classes look similar to Gaussian distribution. In Section 3, we showed the classification result with Gaussian-Bayesian classifier (Moore 2001). However, the performance of GBC classifier is not as good as SMO and logistic regression, which implies that the linear boundary fits this problem better.

2. Results

In total we have 139 textual features. However, some of them didn’t appear in all or most of the documents, thus were not valuable for classification. By eliminating uncommon variables, we have 112 features left. The quality set we selected contains “Depth”, “Multi-side” and “Objectivity”, since it was shown by K.B. Ng that they have the strongest correlations with the textual features. The testing environment is WEKA-3-2-3, 2-fold cross validation. The machine learning methods we chose were: Decision Tree (J48), Naïve Bayes (NB), Support Vector Machine (SMO) and Logistic Regression (LR) (Darlington 1990).

¹ This paper involves other people’s work not published, so I list the people’s names without reference links.

Table 1. The performance table of different methods with 112 textual features (Weka 2 folds). The numbers in each cell are: the percentage of correctly classified document in class I (low scores) / the percentage of correctly classified document in class II (high scores) / and the overall percentage of correctly classified documents \pm 95% confidence interval of overall prediction assuming *t*-distribution. The numbers in the title rows are: the number of samples in class I / number of samples in class II.

	DEPTH (1119/894)	MULTI-SIDE (1038/975)	OBJECTIVITY (995/1018)
J48	62.6/52.6/58.2 \pm 1.9	60.3/56.1/58.3 \pm 1.1	52.1/51.6/51.8 \pm 1.1
NB	81.6/42.4/64.2 \pm 1.0	78.7/43.7/61.8 \pm 1.0	47.3/59.2/53.4 \pm 1.0
SMO	81.5/45.6/65.5 \pm 1.8	78.0/56.6/67.7 \pm 0.7	51.9/61.7/56.8 \pm 0.66
LR	74.4/51.1/64.0 \pm 2.4	70.5/60.8/65.8 \pm 2.8	55.2/59.3/57.3 \pm 2.8

Table 1 shows that all the methods have performance better than chance. Support Vector Machine and Logistic regression have the best performance while SVM is more stable. This gives us the hypothesis that the methods with an adaptive linear boundary will have better performance.

The next question is which features are best for the judgment of each quality. It turned out that we can get predictive accuracy of 68.9/60.5/65.1%, if we only use the document length as the feature in logistic regression and set the classification cutoff to 0.4. And most of other features are proportional to the document length since they are simply the numbers of times each specific feature appears in the document. It shows the strong relationship between the human judgment of the depth and document length. For “Multi-side” and “Objectivity”, the useful features are not so obvious. With an idea from Ying Sun, we ran discriminant analysis (Klecka, 1980) in SPSS 10 times. Each time discriminant analysis gave a set of features that were important to make good predictions, and this set was not very stable. Table 2 shows the ones that appeared most often in the features set.

Table 2. The features appeared most often in discriminant analysis result.

	DEPTH	MULTI-SIDE	OBJECTIVITY
FEATURES	Cd, log_n, n_u_word, persfull, timeunit, n_back, seem	cc, expert, l_bracke, log_n, n_forw, person_u, rbr, rbs,	cc, cd, expert, n_forw, pdt, rbs, say, subtitle

Some of these features are easy to understand, like log_n, which is the logarithm of the document length, while others are not, e.g. l_bracket, which is the number of the left brackets. We know the left bracket usually comes

along with the right bracket; it’s strange that the right brackets are not as useful as the left ones.

To address this problem, data reduction is a natural idea, since it takes care of the correlations between the features. We did PCA factor analysis (Reyment 1993) on the 112 features, and took the first 30 factors. These factors are responsible for 82.9% of the total variance. We tried to predict the qualities with the first 1 factor, the first 5 factors and the first 30 factors, using logistic regression. The results are shown in table 3.

Table 3. The performance of logistic regression (2 folds)

	DEPTH (1119/894)	MULTI-SIDE (1038/975)	OBJECTIVITY (995/1018)
1 FCTR	81.7/42.9/64.4 \pm 0.1	74.7/43.5/59.6 \pm 0.57	27.2/76.5/52.1 \pm 1.9
5 FCTR	83.4/43.9/65.8 \pm 1.6	76.3/56.3/66.6 \pm 1.4	51.5/60.8/56.1 \pm 0.92
30 FCTR	80.1/46.9/65.3 \pm 2.6	74.9/58.3/66.9 \pm 0.76	54.4/62.6/58.5 \pm 3.3

We can see that for “Depth”, a single factor can have predictive accuracy almost as good as that we use all the variables/factors, while “Multi-side” has much better performance with 5 factors, and “Objectivity” needs more than 5 factors.

Table 4 shows the classical machine learning methods with 30 factors. Again, SMO beats other methods in both performance and stability.

Table 4. The performance table of different methods with 30 factors (Weka 2 folds)

	DEPTH (1119/894)	MULTI-SIDE (1038/975)	OBJECTIVITY (995/1018)
J48	68.0/55.5/62.4 \pm 3.7	71.0/52.4/62.0 \pm 2.0	48.8/62.6/55.8 \pm 3.7
NB	82.9/42.3/64.9 \pm 0.49	78.3/41.0/60.2 \pm 1.4	55.2/50.6/52.9 \pm 2.7
SMO	82.5/36.6/64.4 \pm 1.9	84.4/45.6/65.6 \pm 1.9	36.9/78.7/58.0 \pm 2.1

3. Noise Analysis and Application of Gaussian-Bayesian Classifier

After trying all these methods, we started to get an impression that there is an upper limit on the predictive accuracy, for example, we never got a predictive accuracy better than 60% for “Objectivity”. Knowing such an upper limit will be a good indicator of our expectation on performance. We know whether we need to try more classification algorithms, or we need to define new features and collect the data.

Let's look at the methods we used. Except for Naïve Bayes, all other methods try to find an optimal linear boundary (SMO, NN, and Logistic Regression) or multiple linear boundaries (J48 decision tree). What is the optimal classification boundary for noisy data? Suppose we have only one feature to predict the quality, we use a Gaussian distribution to approximate the distribution of this feature in class I and class II, namely, $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$. As shown in figure 1(a), the optimal linear boundary is the intersection point of these two distributions. The performance of this optimal boundary depends on the two distributions, see figure 1(b), if the two distributions are very close, the error rate will be very large (the shadowed area). In other words, this feature is not sufficient for accurate classification.

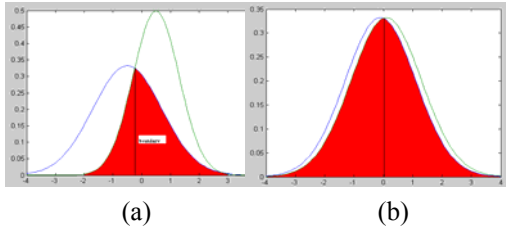


Figure 1. The optimal classification boundary for a single feature. The black straight line in the middle is the optimal boundary, the shadowed area is misclassified. (a) an example (b) if the two distributions are very alike, the error rate could be very large (up to 50%)

Now back to our multi-feature problem. Again, here we used the factor analysis from the previous section. Figure 2 shows the distribution of the two classes over the subspace spanned by the first 2 factors. We can see that although the class II has larger variance than class I, the centers of their mass are mixed tightly.

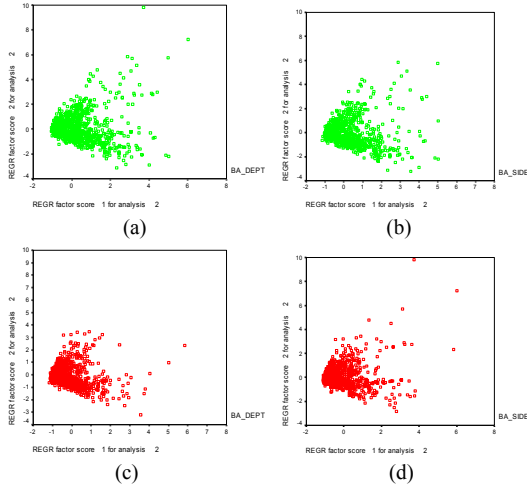


Figure 2. The distribution of class I and class II on the scatter plot of most significant factors. (a) and (c) are the two classes on separate scatter plots. (b), (d) are those for the quality multi-side

To be more precise, we computed the means and standard deviations of the factors for each class separately. Amazingly, although we didn't include any classification information in building factors, the means of the factors in class I have similar amplitude as those in class II, and have different signs. This definitely deserves more attention later. For now, we simply assume that the distributions are Gaussians with the computed means and standard deviations. Table 5 shows the first 3 factors for quality depth.

Now we are ready to compute the optimal boundary for each factor, separately. We just pick the single factor with the best performance as the feature to do prediction. Table 6 shows the result of this method applied to the data.

Table 5. The means and standard deviations of factors for class I and class II, quality "Depth"

DEPTH	FACTOR 1	FACTOR 2	FACTOR 3
MEAN I	-0.283	-0.052	0.013
MEAN II	0.354	0.065	-0.016
STDEV I	0.790	0.775	0.709
STDEV II	1.12	1.22	1.27

Table 6. The performance of single factor optimal boundary. e.g. if the value of factor 1 is greater than 0.375, we classify it as depth class II.

	DEPTH	MULTI-SIDE	OBJECTIVITY
FACTOR	Factor 1	Factor 3	Factor 3
BOUNDARY	0.375	0.25	0.044
CLASS I CLSFD	84.1%	78.0%	61.8%
CLASS II CLSFD	38.6%	43.0%	51.9%
ALL CLASSIFIED	63.8%	61.0%	56.8%

We can see that, with one single feature Gaussian optimal boundary, we can get the overall performance close to the classical machine learning methods.

Encouraged by this fact, we tried Gaussian-Bayesian classifier (Moore 2001) on this problem. Briefly, this classifier can be described as:

if $P(x|C1)P(C1) > P(x|C2)P(C2)$ then classify x as class I else classify x as class II.

$$p(x | C) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)} p(C)$$

In practice, the covariance matrix Σ is singular, thus we had to make some change to the formula. Σ can be transformed into $Q\Lambda Q'$, where Q is orthonormal

eigenvector matrix, and Λ is eigenvalue diagonal matrix. Thus,

$$\begin{aligned} (x - \mu)' \Sigma^{-1} (x - \mu) &= (x - \mu)' Q \Lambda^{-1} Q' (x - \mu) \\ &= (x - \mu)'_k \Sigma_k^{-1} (x - \mu)_k = \sum_{i=1}^k \lambda_i (x - \mu)' (x - \mu) \end{aligned}$$

where k is number of non-zero eigenvalues. In fact, we can also ignore small eigenvalues since they make less contribution to the probability number. Figure 4 shows the performance of GBC with different number of largest eigenvalue/eigenvectors selected.

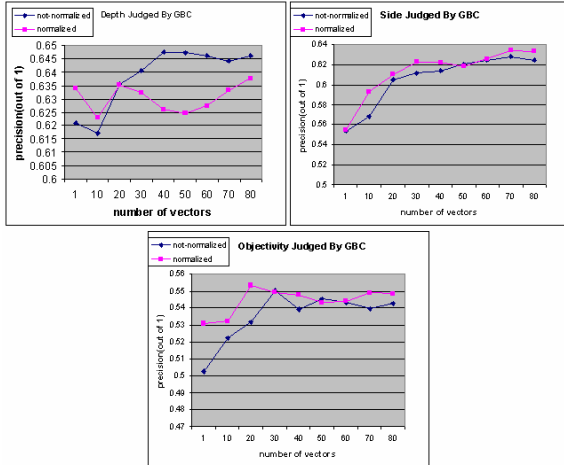


Figure 3. The performance of Gaussian-Bayesian Classifier with different number of eigenvalue/eigenvectors. The upper-left is for quality “depth”, the upper right is for “multi-side”, the bottom one is for “objectivity”. For each graph, the horizontal axis is the number of eigenvalues, vertical axis is the prediction accuracy. Each number is the average of 5 tests. Each graph has two lines, the one of them generated with raw feature numbers, the other is generated with the feature number normalized (divided by document length).

In figure 3, we see that it doesn’t matter very much if we normalize the features or not. Basically, including more important factors of features will give better performance. We also see that the performance we got from Gaussian-Bayesian Classifier is not as good as what we got from logistic regression or SMO. We think this can be described as that the boundary is more close to a linear boundary than a multivariate Gaussian boundary. Actually, we listed the distribution graphs of a feature in figure 4. This distribution is very typical in the features we have. We can see that they are not Gaussian even if we don’t consider the correlation between them.

Specifically, we want to try two things in the future: first is to use more accurate distribution to replace Gaussian distribution when we use Gaussian-Bayesian Classifier; second is to use more than one linear boundary to see whether we can get better performance.

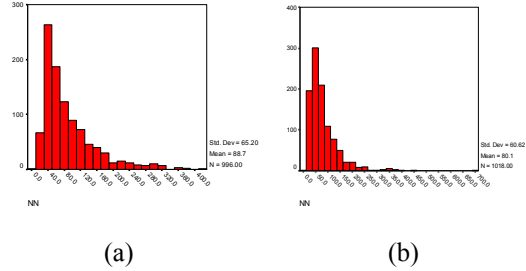


Figure 4. The distributions of feature NN, (a) is the distribution with low objectivity, (b) is the distribution with high objectivity.

Acknowledgement

The study is part of a large-scale multi-institutional project, named HITIQA, supported by the Advanced Research and Development Activity (ARDA)’s Advanced Question Answering for Intelligence (AQUAINT) Program under contract number 2002-H790400-000.

The views expressed in this article are those of the authors, and do not necessarily represent the views of the sponsoring agency. We gratefully acknowledge the contribution of all the participants of the quality experiments at both Albany and Rutgers, especially Tomek Strzalkowski, Paul Kantor, K.B. Ng, Nina Wacholder, Robert Rittman, Peng Song and Ying Sun.

References

Tang, R., Ng, K.B., Strzalkowski, T. & Kantor, P. (2003). *Toward Machine Understanding of Information Quality. Under submission.* Paper submitted.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., & Wilks, Y.(2000). *Experience of Using GATE for NLP R&D.* Workshop on Using Toolsets and Architectures To Build NLP System at COLING-2000, Luxembourg.

Klecka, William, R. (1980) *Discriminant analysis.* Sage University Paper series on quantitative applications in the social sciences, series no. 07-019. Thousand Oaks, CA: Sage.

Darlington, R. B. (1990), *Regression and linear models.* New York: McGraw-Hill. Chapter 18.

Reyment, R. & Joreskog, K.G. (1993) *Applied factor analysis in the natural science.* Cambridge University Press.

Moore, A. (2001) <http://www-2.cs.cmu.edu/~awm/tutorials/gaussbc12.pdf>