Predicting gender from blog posts

Cathy Zhang[*]and Pengyu Zhang[†]

December 10, 2010

1 Introduction

Blogs are informal, personal writings that people post on their own blog sites. Nowadays, blogging is an important online activity. People share blogs with their friends and family members. The topics of blog posting cover almost everything, ranging from personal life, political opinions, recipes, product reviews, or even just random rants. Although some bloggers review their biologically information on their blog site, many don't make such information public. Thereful, automatically classifying the gender or age of a blog author may have important applications in many commercial domains, such as targeted advertising and product development [8]. In blog search, this information may help people find blogs that they are interested in. From a research perspective, blog author gender classification is also an interesting problem. Blog posts are typically short and unstructured. They differ tremendously from formal texts, since they may have informal sentences, grammer errors, slang words and phrases, and wrong spellings. These characteristics of blog posts may complicate any classification or categorization attempts.

The goal of our project is to identify author gender of blogs coming from a wide variety of sources. We are interested in knowing how well we can tackle this problem, what methods and features are most effective in this task.

2 Literature Review

There are two recent works that attempts to tackle the same task. [11] used naive Bayes classifer on word features (with stop words removed) and ”non-traditional” features like background color, word fonts and cases, punctuation marks, and emoticons. [8] tried naive Bayes, SVM, and SVM regression as classifiers. In addition to words, they used many other features derived from the text. In particular, they introduced a POS sequence feature selector and an ensemble feature selection algorithm. The majority of this work is based on their attempts.

[*] zhang20t@mtholyoke.edu

[†] pyzhang@cs.umass.edu

Blog author gender classification is one kind of text classification problem, and there are previous work on feature selection and classification algorithms. [13] and [4] are two comprehensive studies that compared feature selection metrics on text classification. [2] and [6] introduced respectively two approaches to feature selection, the filtering and wrapper approaches. [12] compares five classifiers: SVM, kNN, naive Bayes, a nerual network approach(NNet), and least squares Fit (LLSF) on text categorization. [7] proposed using string kernel with SVM for text classifier.

There are a lot of empirical studies devoted to blog mining or gender specific text analysis. [1] explores how age and gender affect writing style and topic using texts from blogosphere, and they found significant stylistic and content-based indicators. [3] investigated authorship gender mining from e-mail text documents. Recently [10] anaylzed co-occurance of slang words to help predicting age and gender.

3 Blog Data Set

We use the same data set as [8]. This data set includes blog posts from many blog hosting sites and blog search engines, e.g., blogger.com, technorati.com, etc. The data set consists of 3226 blog entries, each with a gender label. Out of the 3226 posts, 1551 (48.08%) were written by men and 1672 (51.92%) were written by women. The average length of these blogs is 422 words. The topics included are truly diverse and general.

3.1 Features

As in most text classification tasks, words are the most important features. Here we also consider features that are suggested in similar previous work.

3.1.1 Words

The main feature here are the words and punctuations that people uses. We use the tokenizer that comes with the part of speech(POS) tagger we use [1]. There are 56024 unique words and punctuations. We can either use binary representation, i.e. whether a word exist in the post or not, or term frequency as the feature. We tried both, but it seems that binary representation is more effective in all cases.

We also considered stemming our words using a porter stemmer[2]. However, it did not improve classification accuracy.

3.1.2 Average word/sentence length

The simplest additional feature we can think of is the average word length and average sentence length. Note that since bloggers have different styles, it is very hard to determine what is a sentence. Some posts have no punctuations at all. We take

[1] `http://pypi.python.org/pypi/topia.termextract/`

[2] `http://tartarus.org/~martin/PorterStemmer/python.txt`

either natual sentence end punctuation, e.g. ".", "?", "!", etc. and new line as indication of end of sentence. Table 1 shows the statistics of the feature.

Table 1: Average word/sentence length

Mean (Std.)	Avg. word length	Avg. sentence length
Male	4.3559(.1746)	19.0883(8.9095)
Female	4.2130(.3825)	17.8864(7.7671)

3.1.3 POS tags

Part of speech analysis is important in processng texts. [8] uses a metric called "F-measure" as a feature. "F-measure" is based on the frequency of POS usage in a text. It was first proposed in [5] and used in [9]. [1] finds out that *articles* and *prepositions* are used significantly more by male bloggers, while *personal pronouns*, *conjunctions*, and *auxiliary verbs* are used significantly more by female bloggers.

We use a POS tagger [3] to tag each token. The tagger has too many types of tags. For simplicity, we only keep track of the most common types of tags: "NN"(noun), "VB"(verb), "JJ"(adjective), "PRP"(personal pronoun), "UH"(interjection), and "RB"(adverb). We uses percentage instead of simple count for each tag because it seems to work better in our experiments. Here we also give the basic statistics of this feature in table 2.

Table 2: POS tags

Mean (Std.)	Male	Female
NN	44.54%(10.14%)	40.64%(9.78%)
VB	23.57%(5.31%)	24.47%(5.33%)
JJ	10.57%(3.81%)	10.35%(3.97%)
PRP	13.34%(5.89%)	16.09%(6.09%)
UH	0.23%(0.56%)	0.25%(0.55%)
RB	7.76%(3.30%)	8.20%(3.42%)

[8] also suggested keeping track of POS sequences. We tried using POS 1,2,3-grams, but they did not seem to improve classification accuracy.

3.1.4 Word factor analysis

Word factor analysis refers to the process of finding groups of similar works that tend to occur in similar documents. We use again the finds in [1]. There are 20 word lists (with suggested labels for reference). Here are 3 example lists:

1. **Conversation** *know, people, think, person, tell, feel, friends, talk, new, talking, mean, ask, understand, feelings, care, thinking, friend, relationship, realize, question, answer, saying*

[3] `http://pypi.python.org/pypi/topia.termextract`

2. **AtHome** *woke, home, sleep, today, eat, tired, wake, watch, watched, dinner, ate, bed, day, house, tv, early, boring, yesterday, watching, sit*

3. **Family** *years, family, mother, children, father, kids, parents, old, year, child, son, married, sister, dad, brother, moved, age, young, months, three, wife, living, college, four, high, five, died, six, baby, boy, spend, christmas*

4 Methods

4.1 Feature selection algorithms

Dimension reduction or feature selection is very important in this classification task, since we have many more features than observations. Among the features, there are a lot of noises that may harm classification. For example, intuitively, ”stop words,” i.e. common words like ”and,” ”to,” etc., may give little insight into author gender. Generic dimension reduction technique, such as PCA cannot be applied here since our feature space is too big. So, instead we follow the suggestion from [8].

[8] mentioned two common approaches to feature selection: the *filter* [2] and the *wrapper* [6] approaches. In the filtering approach, features are ranked by some metric and only the top k features are retained for classification. The wrapper approach adds new features into the existing set if the new features improve the classification accuracy. Here, we only use the filtering approach and picked 3 selection criteria: information gain, mutual information, and χ^2 statistic. While implementing our algorithm, we also referenced [13] and [4] for comprehensive studies of feature selection metrics in text categorization.

Let $C = \{c, \bar{c}\}$ be the two gender classes, and $F = \{f_1, f_2, ..f_n\}$ the set of features. We describe the formulation of each feature selection criterion below.

4.1.1 Information gain (IG)

Information gain is frequently employed as a term-goodness critierion in the field of machine learning. It measures the number of bits of information obtained for category prediction by knowing the presence or absense of a term in a document [13]. Note that information gain is a entroy based metric.

$$IG(f) = -\sum_{c,\bar{c}} P(c)\log P(c) + \sum_{f,\bar{f}} P(f) \sum_{c,\bar{c}} P(c|f)\log P(c|f) \tag{1}$$

4.1.2 Mutual information (MI)

Mutual information is commonly used in statistical language modeling [8] [13]. The mutual information $MI(f, c)$ between a class c and a feature f is defined as:

$$MI(f,c) = \sum_{f,\bar{f}} \sum_{c,\bar{c}} P(f,c) log \frac{P(f,c)}{P(f)P(c)} \tag{2}$$

Note that since we only have two classes, $MI(f,c) = MI(f,\bar{c})$. So we just denote equation 2 as $MI(f)$.

4.1.3 χ^2 statistic

The χ^2 statistic measures the lack of independence between a feature f and class c, and can be compared to the χ^2 distribution with one degree of freedom [8] [13]. Let the 2×2 contigency table of a feature f and a class c to be the following.

Table 3: Two-way contingency table of f and c

	c	$\bar{c}$
f	W	X
$\overline{f}$	Y	Z

Here, W denotes the number of documents in the corpus in which feature f and class c co-occur. Let $N = W + X + Y + Z$. The χ^2 test is defined as:

$$\chi^2(f,c) = \frac{N(WZ - YX)^2}{(W+Y)(X+Z)(W+X)(Y+Z)} \tag{3}$$

Similar to MI, since we only have 2 classes, $\chi^2(f,c) = \chi^2(f,\bar{c})$. So we denote equation 3 as $\chi^2(f)$.

4.2 Classification algorithms

4.2.1 Naive Bayes

Naive Bayes is the first approach we tried. Let $C = (c_1, c_2)$ be the gender class, and $F = (f_1, f_2, ...f_n)$ are features, according to Bayes theorem:

$$P(c|F) = \frac{P(c)P(F|c)}{P(F)}$$

The naive Bayes assumption is that:

$$\hat{P}(F|c) = \prod_{i=1}^{n} \hat{P}(f_i|c)$$

So the clasification algorithm is:

$$\arg\max_c P(C = c|F) = \arg\max_c P(C = c) \prod_i^n P(f = f_i|C = c)$$

4.2.2 SVM

SVM is proven to be effective in many machine learning applications. We use SVM with different kernel: linear kernel, polynomial kernel, rbf kernel, and string kernel. We tried different parameters for each kernel so that we can get the best result. We use the libsvm package [4] and svm light package. It implements linear, rbf, and polynomial kernels and allows self defined kernel function. It also automatically run cross validation. We also take a string kernel implementation [5] and plug it into libsvm. Unlike the other classifcation algorithms here that take feature vectors, we use the original texts as feature. The string kernel function computes edit distance up to length k substrings, so it is computationally expensive to run. We therefore chop blog posts into short segments: 15 tokens per segment, and we can only work with 10000 segments.

4.2.3 LDA

We also use linear discriminant analysis (LDA). LDA finds a linear combination of features which characterize or separate two or more classes. It seeks to minimize the variance within each class while maximizing the variance across classes. Since it is too inefficient to run LDA on the raw data set, we have to apply feature selection methods to get a reduced set of features first, and then run LDA on top of the reduced set of features.

5 Results and Discussions

5.1 Features

This experiment tries to test the performance of naive Bayes classifier on different feature types. 2800 blogs are chosen from 3226 blogs as train data set. The rest 426 blogs are used as test set. The results of naive bayes based classifier are shown in Table.4. The columns stand for different data feature types.

Table 4: Naive bayes classifier

Raw data	Stemmed	Term frequency (TF)	Stemmed and TF	IG	IG and TF
55.09%	53.7%	51.39%	51.39%	66.67%	62.44%

Naive bayes classifier based on raw blog data set can achieve accuracy of 55.09%. Yet adding stemmer and term frequency reduces prediction accuracy. Supprisingly, the accuracy of word frequency is not as good as raw data. The reason may be the high frequency of gender non-sensitive word that cause much noise in classification. Information gain increases the accuracy greatly and term frequency reduces accuracy. So we can get the conclusion that stemmed text and term frequency is not important for gender classification.

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[5] http://www.learning-kernel-classifiers.org/code/string_kernels/strings.c

5.2 Classification algorithms

5.2.1 SVM

We use SVM light application to learn different SVM models. In this experiment, SVM is run on raw data set with linear kernel, rbf kernel and polynomial kernel. The results can be seen in Table.5. We can see that the performance of linear kernel is the best. However, the accuracy of SVM is not high compared with naive bayes classifier. Besides, SVM with rbf kernel is tested with different rbf kernel parameters γ. If γ is large, the accuracy of SVM based on rbf kernel decreases much.

Table 5: SVM

	Kernel parameters	Accuracy
Linear kernel		64.35%
rbf kernel	$\gamma = 0.5$	50.93%
	$\gamma = 0.01$	59.26%
	$\gamma = 0.001$	62.96%
	$\gamma = 0.0001$	62.73%
Polynomial kernel	$d = 1$	62.96%
	$d = 2$	57.64%

String kernel does not work well either. In the one experiment we did (condition given in the methods section) by 5 fold cross-validation, it gives 55.5922% accuracy. It may be because we are only using a naive implementation.

5.2.2 LDA

LDA in MATLAB cannot support 56024 features. As most of the features are noise for classification, we use information gain to select top features and apply LDA on it. The results is shown in Table.6.

Table 6: LDA with information gain

Selected feature number	Accuracy
50	61.03%
100	60.56%
200	63.62%
400	63.38%
600	62.91%
800	64.55%
1000	63.38%

5.3 Feature selection

5.3.1 Comparison of feature selection criteria

Table 7 shows the accuracy of SVM (linear kernel) using different feature selection methods. We run the experiment under 10 fold cross-validation on the entire data set. Rows are results from retaining different number of features from the original feature vector. Columns compare the results using different feature selection methods: the columns without * shows results using only binary word features; the columns with * shows results using binary word features and additional features described in section 3.1.

Table 7: Accuracy of SVM linear with different feature selection methods

	IG	IG*	χ^2	χ^{2*}	MI	MI*
200	69.2808%	69.4978%	68.2579%	68.5059%	62.8642%	63.5152%
500	69.9938%	70.7378%	70.5208%	70.4588%	64.7861%	64.7241%
1000	71.4507%	72.1017%	70.4588%	70.6758%	65.8091%	64.8791%
2000	70.9547%	71.3887%	70.0248%	70.4278%	63.9802%	63.6702%
5000	70.9237%	71.3887%	70.8308%	71.0477%	65.3441%	65.4681%
10000	71.9467%	71.5437%	70.8617%	71.4197%	66.305%	66.367%
20000	70.6758%	71.0167%	70.0558%	70.7068%	67.142%	67.359%
56024	67.235%	67.049%	67.235%	67.049%	67.235%	67.049%

Here, we see that IG and χ^2 both significantly improve classification accuracy. Their improvements are comparable, with IG slightly better. MI actually negatively affects the prediction accuracy. Note that this result is consistent with the observation made by [13]. Specifically, they found that IG and χ^2 are the most effective methods in their experiments. Moreover, they suspect that MI has poor performance due to its bias towards favoring rare terms, and its sensitivity to probability estimation errors.

Also, additional features slightly improves the results with IG and χ^2 in general, but not with MI.

5.3.2 Accuracy vs. number of features

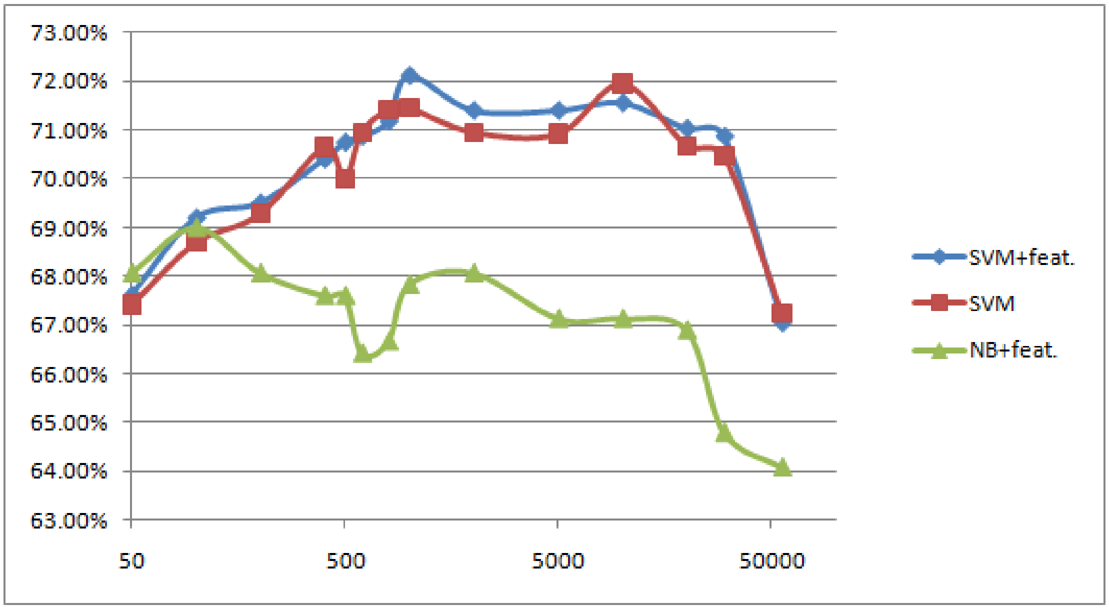

Figure 1: Predicting accuracy v. number of features

Figure 1 shows the prediction accuracy vs. no. of features using SVM/NB+IG with or without additional features. The data is collected from the previous experiment. We see that naive Bayes matches the accuracy of SVM with a small number of features. SVM seems to operate the best in the range of 1000˜10,000 features.

6 Conclusions

For the blog author gender classification task, we found that the best prediction accuracy we could achieve is 72.1017%. This resulted is achieved using all features mentioned, IG as feature selection criterion, and SVM (linear kernel) as the classifier. We found that binary word feature is in general more effective than term frequency. Additional features slightly improves prediction accuracy in combination with feature selection mechanisms. IG and χ^2 are two feature selection criteria that works well in selecting words carry the most discriminatory power against other terms. SVM (linear kernel) is the best classifier for this task.

This is obviously our first attempt at a complex text classification task. We could not reproduce the best result as in [8], since we do not have enough time to implement their complex feature selection algorithm and their POS sequence selector. We found that feature selection is very important, because using all terms in the corpos introduce too many noises. Also, linguistics studies, cognitive science studies, and other empirical studies on gender preferential terms, part-of-speech analysis, etc. are very helpful in deciding what additional features can be useful.

References

[1] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler. Mining the blogosphere: Age, gender and the varieties of self-expression. *First Monday*, (9), 2007.

[2] Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[3] Malcolm Corney, Olivier de Vel, Alison Anderson, and George Mohay. Gender-preferential text mining of e-mail discourse. In *In 18th Annual Computer Security Applications Conference. 2002*, pages 21–27, 2002.

[4] George Forman, Isabelle Guyon, and Andr Elisseeff. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[5] F. Heylighen and J. Dewaele. Variation in the contextuality of language: an empirical measure. *Foundations of Science*, pages 293–340, 2002.

[6] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997.

[7] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels, 2002.

[8] A. Mukherjee and B. Liu. Improving Gender Classification of Blog Authors. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010.

[9] S. Nowson, J. Oberlander, and A. J. Gill. Gender, genres, and individual differences. In *Proceedings of the 27th annual meeting of the Cognitive Science Society*, pages 1666–1671, 2005.

[10] R. Rajendra Prasath. Learning age and gender using co-occurrence of non-dictionary words from stylistic variations. In *Proceedings of the 7th international conference on Rough sets and current trends in computing*, RSCTC'10, pages 544–550, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] X. Yan and L. Yan. Gender classification of weblog authors. In *AAAI*, 2006.

[12] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 42–49, New York, NY, USA, 1999. ACM.

[13] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, pages 412–420, 1997.