Improving Gender Classification of Blog Authors

Arjun Mukherjee
Department of Computer Science
University of Illinois at Chicago
851 South Morgan Street
Chicago, IL 60607, USA
amukherj@cs.uic.edu

Bing Liu
Department of Computer Science
University of Illinois at Chicago
851 South Morgan Street
Chicago, IL 60607, USA
liub@cs.uic.edu

Abstract

The problem of automatically classifying the gender of a blog author has important applications in many commercial domains. Existing systems mainly use features such as words, word classes, and POS (part-of-speech) n-grams, for classification learning. In this paper, we propose two new techniques to improve the current result. The first technique introduces a new class of features which are variable length POS sequence patterns mined from the training data using a sequence pattern mining algorithm. The second technique is a new feature selection method which is based on an ensemble of several feature selection criteria and approaches. Empirical evaluation using a real-life blog data set shows that these two techniques improve the classification accuracy of the current state-of-the-art methods significantly.

1 Introduction

Weblogs, commonly known as blogs, refer to online personal diaries which generally contain informal writings. With the rapid growth of blogs, their value as an important source of information is increasing. A large amount of research work has been devoted to blogs in the natural language processing (NLP) and other communities. There are also many commercial companies that exploit information in blogs to provide value-added services, e.g., blog search, blog topic tracking, and sentiment analysis of people's opinions on products and services. Gender classification of blog authors is one such study, which also has many commercial applications. For example, it can help the user find what topics or products are most talked about by males and females, and what products and services are liked or disliked by men and women. Knowing this information is crucial for market intelligence because the information can be exploited in targeted advertising and also product development.

In the past few years, several authors have studied the problem of gender classification in the natural language processing and linguistic communities. However, most existing works deal with formal writings, e.g., essays of people, the Reuters news corpus and the British National Corpus (BNC). Blog posts differ from such text in many ways. For instance, blog posts are typically short and unstructured, and consist of mostly informal sentences, which can contain spurious information and are full of grammar errors, abbreviations, slang words and phrases, and wrong spellings. Due to these reasons, gender classification of blog posts is a harder problem than gender classification of traditional formal text.

Recent work has also attempted gender classification of blog authors using features such as content words, dictionary based content analysis results, POS (part-of-speech) tags and feature selection along with a supervised learning algorithm (Schler et al., 2006; Argamon et al., 2007; Yan and Yan, 2006). This paper improves these existing methods by proposing two novel techniques. The first technique adds a new class of pattern based features to learning, which are not used in any existing work. The patterns are frequent sequences of POS tags which can capture complex stylistic characteristics of male and female authors. We note that these patterns are very different from the traditional n-grams because the

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 207–217,
MIT, Massachusetts, USA, 9-11 October 2010.

patterns are of variable lengths and need to satisfy some criteria in order for them to represent significant regularities. We will discuss them in detail in Section 3.5.

The second technique is a new feature selection algorithm which uses an ensemble of feature selection criteria and methods. It is well known that each individual feature selection criterion and method can be biased and tends to favor certain types of features. A combination of them should be able to capture the most useful or discriminative features.

Our experimental results based on a real life blog data set collected from a large number of blog hosting sites show that the two new techniques enable classification algorithms to significantly improve the accuracy of the current state-of-the-art techniques (Argamon et al., 2007; Schler et al., 2006; Yan and Yan, 2006). We also compare with two publicly available systems, *Gender Genie* (BookBlog, 2007) and *Gender Guesser* (Krawetz, 2006). Both systems implemented variations of the method given in (Argamon et al., 2003). Here, the improvement of our techniques is even greater.

2 Related Work

There have been several recent papers on gender classification of blogs (e.g., Schler et al., 2006, Argamon et al., 2007; Yan and Yan, 2006; Nowson et al., 2005). These systems use function/content words, POS tag features, word classes (Schler et al., 2006), content word classes (Argamon et al., 2007), results of dictionary based content analysis, POS unigram (Yan and Yan, 2006), and personality types (Nowson et al., 2005) to capture stylistic behavior of authors' writings for classifying gender. (Koppel et al. 2002) also used POS n-grams together with content words on the British National Corpus (BNC). (Houvardas and Stamatatos, 2006) even applied character (rather than word or tag) n-grams to capture stylistic features for authorship classification of news articles in Reuters.

However, these works use only one or a subset of the classes of features. None of them uses all features for classification learning. Given the complexity of blog posts, it makes sense to apply all classes of features jointly in order to classify genders. Moreover, having many feature classes is very useful as they provide features with varied granularities and diversities. However, this also results in a huge number of features and many of them are redundant and may obscure classification. Feature selection is thus needed. Following the idea, this paper proposes a new ensemble feature selection method which is capable of extracting good features from different feature classes using multiple criteria.

We also note some less relevant literature. For example, (Tannen, 1990) deals with gender differences in "conversational style" and in "formal written essays", and (Gefen and Straub, 1997) reports differences in perception of males and females in the use of emails.

Our new POS pattern features are related to POS n-grams used in (Koppel et al., 2002; Argamon et al., 2007), which considered POS 3-grams, 2-grams and unigrams as features. As shown in (Baayen et. al. 1996), POS n-grams are very effective in capturing the fine-grained stylistic and heavier syntactic information. In this work, we go further by finding POS sequence patterns. As discussed in the introduction, our patterns are entirely different from POS n-grams. First of all, they are of variable lengths depending on whatever lengths can catch the regularities. They also need to satisfy some constraints to ensure that they truly represent some significant regularity of male or female writings. Furthermore, our POS sequence patterns can take care of n-grams and capture additional sequence regularities. These automatically mined pattern features are thus more discriminating for classification.

3 Feature Engineering and Mining

There are different *classes* of features that have been experimented for gender classification, e.g., F-measure, stylistic features, gender preferential features, factor analysis and word classes (Nowson et al., 2005; Schler et al., 2006; Corney et al., 2002; Argamon et al., 2007). We use all these existing features and also propose a new class of features that are POS sequence patterns, which replace existing POS n-grams. Also, as mentioned before, using all feature classes gives us features with varied granularities. Upon extracting all these classes of features, a new *ensemble feature selection* (EFS) algorithm is proposed to select a subset of good or discriminative features.

Below, we first introduce the existing features, and then present the proposed class of new pattern based features and how to discover them.

3.1 F-measure

The F-measure feature was originally proposed in (Heylighen and Dewaele, 2002) and has been used in (Nowson et al., 2005) with good results. Note that F-measure here is not the F-score or F-measure used in text classification or information retrieval for measuring the classification or retrieval effectiveness (or accuracy).

F-measure explores the notion of implicitness of text and is a unitary measure of text's relative contextuality (implicitness), as opposed to its formality (explicitness). Contextuality and formality can be captured by certain parts of speech. A lower score of F-measure indicates contextuality, marked by greater relative use of pronouns, verbs, adverbs, and interjections; a higher score of F-measure indicates formality, represented by greater use of nouns, adjectives, prepositions, and articles. F-measure is defined based on the frequency of the POS usage in a text (*freq.x* below means the frequency of the part-of-speech *x*):

$$F = 0.5 * [(freq.noun + freq.adj + freq.prep + freq.art) - (freq.pron + freq.verb + freq.adv + freq.int) + 100]$$

(Heylighen and Dewaele, 2002) applied the F-measure to a corpus with known author genders and found a distinct difference between the sexes. Females scored lower preferring a more contextual style while males scored higher preferring a more formal style. F-measure values for male and female writings reported in (Nowson et al., 2005) also demonstrated a similar trend. In our work, we also use F-measure as one of the features.

3.2 Stylistic Features

These are features which capture people's writing styles. The style of writing is typically captured by three types of features: part of speech, words, and in the blog context, words such as *lol*, *hmm*, and *smiley* that appear with high frequency. In this work, we use words and blog words as stylistic features. Part of speech features are mined using our POS sequence pattern mining algorithm. POS n-grams can also be used as features. However, since we mine all POS sequence patterns and use them as features, most discriminative POS n-grams are already covered. In Section 5, we will also show that POS n-grams do not perform as well as our POS sequence patterns.

3.3 Gender Preferential Features

Gender preferential features consist of a set of signals that has been used in an email gender classification task (Corney et al., 2002). These features come from various studies that have been undertaken on the issue of gender and language use (Schiffman, 2002). It was suggested by these studies and also various other works that women's language makes more frequent use of emotionally intensive adverbs and adjectives like "so", "terribly", "awfully", "dreadfully" and women's language is more punctuated. On the other hand, men's conversational patterns express "independence" (Corney et al., 2002). In brief, the language expressed by males is more proactive at solving problems while the language used by females is more reactive to the contribution of others - agreeing, understanding and supporting. We used the gender preferential features listed in Table 1, which indicate adjectives and adverbs based on the presence of suffixes and apologies as used in (Corney et al., 2002). The feature value assignment will be discussed in Section 5.

f1	words ending with *able*
f2	words ending with *al*
f3	words ending with *ful*
f4	words ending with *ible*
f5	words ending with *ic*
f6	words ending with *ive*
f7	words ending with *less*
f8	words ending with *ly*
f9	words ending with *ous*
f10	*sorry* words

Table 1: Gender preferential features

3.4 Factor Analysis and Word Classes

Factor or word factor analysis refers to the process of finding groups of similar words that tend to occur in similar documents. This process is referred to as meaning extraction in (Chung and Pennebaker, 2007). Word lists for twenty factors, along with suggested labels/headings (for reference) were used as features in (Argamon et al., 2007). Here we list some of those features (word

classes) in Table 2. For the detailed list of such word classes, the reader is referred to (Argamon et al., 2007). We also used these word classes as features in our work. In addition, we added three more new word classes implying positive, negative and emotional connotations and used them as features in our experiments. These are listed in Table 3.

Factor	*Words*
Conversation	*know, people, think, person, tell, feel, friends, talk, new, talking, mean, ask, understand, feelings, care, thinking, friend, relationship, realize, question, answer, saying*
Home	*woke, home, sleep, today, eat, tired, wake, watch, watched, dinner, ate, bed, day, house, tv, early, boring, yesterday, watching, sit*
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*
Food / Clothes	*food, eating, weight, lunch, water, hair, life, white, wearing, color, ice, red, fat, body, black, clothes, hot, drink, wear, blue, minutes, shirt, green, coffee, total, store, shopping*
Romance	*forget, forever, remember, gone, true, face, spent, times, love, cry, hurt, wish, loved*

Table 2: Words in factors

Positive	*absolutely, abundance, ace, active, admirable, adore, agree, amazing, appealing, attraction, bargain, beaming, beautiful, best, better, boost, breakthrough, breeze, brilliant, brimming, charming, clean, clear, colorful, compliment, confidence, cool, courteous, cuddly, dazzling, delicious, delightful, dynamic, easy, ecstatic, efficient, enhance, enjoy, enormous, excellent, exotic, expert, exquisite, flair, free, generous, genius, great, graceful, heavenly, ideal, immaculate, impressive, incredible, inspire, luxurious, outstanding, royal, speed, splendid, spectacular, superb, sweet, sure, supreme, terrific, treat, treasure, ultra, unbeatable, ultimate, unique, wow, zest*
Negative	*wrong, stupid, bad, evil, dumb, foolish, grotesque, harm, fear, horrible, idiot, lame, mean, poor, heinous, hideous, deficient, petty, awful, hopeless, fool, risk, immoral, risky, spoil, spoiled, malign, vicious, wicked, fright, ugly, atrocious, moron, hate, spiteful, meager, malicious, lacking*
Emotion	*aggressive, alienated, angry, annoyed, anxious, careful, cautious, confused, curious, depressed, determined, disappointed, discouraged, disgusted, ecstatic, embarrassed, enthusiastic, envious, excited, exhausted, frightened, frustrated, guilty, happy, helpless, hopeful, hostile, humiliated, hurt, hysterical, innocent, interested, jealous, lonely, mischievous, miserable, optimistic, paranoid, peaceful, proud, puzzled, regretful, relieved, sad, satisfied, shocked, shy, sorry, surprised, suspicious, thoughtful, undecided, withdrawn*

Table 3: Words implying positive, negative and emotional connotations

3.5 Proposed POS Sequence Pattern Features

We now present the proposed POS sequence pattern features and the mining algorithm. This results in a new feature class. A *POS sequence pattern* is a sequence of consecutive POS tags that satisfy some constraints (discussed below). We used (Tsuruoka and Tsujii, 2005) as our POS tagger.

As shown in (Baayen et. al., 1996), POS n-grams are good at capturing the heavy stylistic and syntactic information. Instead of using all such n-grams, we want to discover all those patterns that represent true regularities, and we also want to have flexible lengths (not fixed lengths as in n-grams). POS sequence patterns serve these purposes. Its mining algorithm mines all such patterns that satisfy the user-specified minimum support (*minsup*) and minimum adherence (*minadherence*) thresholds or constraints. These thresholds ensure that the mined patterns represent significant regularities.

The main idea of the algorithm is to perform a level-wise search for such patterns, which are POS sequences with minsup and minadherence. The *support* of a pattern is simply the proportion of documents that contain the pattern. If a pattern appears too few times, it is probably spurious. A sequence is called a *frequent sequence* if it satisfies minsup. The *adherence* of a pattern is measured using the *symmetrical conditional probability* (SCP) given in (Silva et al., 1999). The SCP of a sequence with two elements $|xy|$ is the product of the conditional probability of each given the other,

$$SCP(x,y) = P(x \mid y)P(y \mid x) = \frac{P(x,y)^2}{P(x)P(y)}$$

Given a consecutive sequence of POS tags $|x_1 \ldots x_n|$, called a *POS sequence* of length n, a dispersion point defines two subparts of the sequence. A sequence of length n contains n-1 possible dispersion points. The SCP of the sequence $|x_1 \ldots x_n|$ given the dispersion point (denoted by *) $|x_1 \ldots x_{n-1} * x_n|$ is:

$$SCP((x_1 \ldots x_{n-1}), x_n) = \frac{P(x_1 \ldots x_n)^2}{P(x_1 \ldots x_{n-1})P(x_n)}$$

The SCP measure can be extended so that all possible dispersion points are accounted for.

Hence the *fair*SCP of the sequence $|x_1 ... x_n|$ is given by:

$$fairSCP(x_1 ... x_n) = \frac{P(x_1 ... x_n)^2}{\frac{1}{n-1}\sum_{i=1}^{n-1} P(x_1 ... x_i) P(x_{i+1} ... x_n)}$$

*fair*SCP measures the adherence strength of POS tags in a sequence. The higher the *fair*SCP value, the more dominant is the sequence. Our POS sequence pattern mining algorithm is given below.

Input: Corpus D = {d | d is a document containing a sequence of POS tags}, Tagset T = {t | t is a POS tag}, and the user specified minimum support (*minsup*) and minimum adherence (*minadherence*).

Output: All POS sequence patterns (stored in *SP*) mined from D that satisfy *minsup* and *minadherence*.

Algorithm mine-POS-pats(D, T, *minsup*, *minadherence*)

1. $C_1 \leftarrow$ count each t ($\in T$) in D;
2. $F_1 \leftarrow \{f \mid f \in C_1, f.\text{count} / n \geq minsup\}$; // $n = |D|$
3. $SP_1 \leftarrow F_1$;
4. for ($k = 2$; $k \leq$ MAX-length; k++)
5. C_k = candidate-gen(F_{k-1});
6. for each document $d \in D$
7. for each candidate POS sequence $c \in C_k$
8. if (c is contained in d)
9. c.count++;
10. endfor
11. endfor
12. $F_k \leftarrow \{c \in C_k \mid \text{c.count} / n \geq minsup\}$;
13 $SP_k \leftarrow \{f \in F_k \mid fair\text{SCP}(f) \geq minadherence\}$
14. endfor
15. return $SP \leftarrow \bigcup_k SP_k$;

Function candidate-gen(F_{k-1})

1. $C_k \leftarrow \varnothing$;
2. for each POS n-gram $c \in F_{k-1}$
3. for each $t \in T$
4. $c' \leftarrow$ addsuffix(c, t); // adds tag t to c as suffix
5. add c' to C_k;
6. endfor
7. endfor

We now briefly explain the mine-POS-pats algorithm. The algorithm is based on level-wise search. It generates all POS patterns by making multiple passes over data. In the first pass, it counts the support of individual POS tags and determines which of them have minsup (line 2). Multiple occurrences of a tag in a document are counted only once. Those in F_1 are called *length 1 frequent sequences*. All length 1 sequence patterns are stored in SP_1. Since adherence is not defined for a single element, we have $SP_1 = F_1$ (line 3). In each subsequent pass k until MAX-length (which is the maximum length limit of the mined patterns), there are three steps:

1. Using F_{k-1} (frequent sequences found in the (k-1) pass) as a set of seeds, the algorithm applies candidate-gen() to generate all possibly frequent POS k-sequences (sequences of length k) (line 5). Those infrequent sequences (which are not in F_{k-1}) are discarded as adding more POS tags will not make them frequent based on the downward closure property in (Agrawal and Srikant, 1994).
2. D is then scanned to compute the actual support count of each candidate in C_k (lines 6-11).
3. At the end of each scan, it determines which candidate sequences have minsup and minadherence (lines 12 - 13). We compute F_k and SP_k separately because adherence does not have the downward closure property as the support.

Finally, the algorithm returns the set of all sequence patterns (line 15) that meet the minsup and minadherence thresholds.

The candidate-gen() function generates all possibly frequent k-sequences by adding each POS tag t to c as suffix. c is a k-1-sequence in F_{k-1}.

In our experiments, we used MAX-length = 7, *minsup* = 30%, and *minadherence* = 20% to mine all POS sequence patterns. All the mined patterns are used as features.

Finally, it is worthwhile to note that mine-POS-pat is very similar to the well-known GSP algorithm (Srikant and Agrawal, 1996). Likewise, it has linear scale up with data size. If needed, one can use MapReduce (Dean and Ghemawat, 2004) with suitable modifications in mine-POS-pats to speed things up by distributing to multiple machines for large corpora. Moreover, mining is a part of preprocessing of the algorithm and its complexity does not affect the final prediction, as it will be later shown that for model building and prediction, standard machine learning methods are used.

4 Ensemble Feature Selection

Since all classes of features discussed in Section 3 are useful, we want to employ all of them. This results in a huge number of features. Many of

them are redundant and even harmful. Feature selection thus becomes important. There are two common approaches to feature selection: the *filter* and the *wrapper* approaches (Blum and Langley, 1997; Kohavi and John, 1997). In the filter approach, features are first ranked based on a feature selection criterion such as information gain, chi-square (χ^2) test, and mutual information. A set of top ranked features are selected. On the contrary, the wrapper model chooses features and adds to the current feature pool based on whether the new features improve the classification accuracy.

Both these approaches have drawbacks. While the wrapper approach becomes very time consuming and impractical when the number of features is large as each feature is tested by building a new classifier. The filter approach often uses only one feature selection criterion (e.g., information gain, chi-square, or mutual information). Due to the bias of each criterion, using only a single one may result in missing out some good features which can rank high based on another criterion. In this work, we developed a novel feature selection method that uses multiple criteria, and combines both the wrapper and the filter approaches. Our method is called *ensemble feature selection* (EFS).

4.1 EFS Algorithm

EFS takes the best of both worlds. It first uses a number of feature selection criteria to rank the features following the filter model. Upon ranking, the algorithm generates some candidate feature subsets which are used to find the final feature set based on classification accuracy using the wrapper model. Since our framework generates much fewer candidate feature subsets than the total number of features, using wrapper model with candidate feature sets is scalable. Also, since the algorithm generates candidate feature sets using multiple criteria and all feature classes jointly, it is able to capture most of those features which are discriminating. We now detail our EFS algorithm.

The algorithm takes as input, a set of n features $F = \{f_1, \ldots, f_n\}$, a set of t feature selection criteria $\Theta = \{\theta_1, \ldots, \theta_t\}$, a set of t thresholds $T = \{\tau_1, \ldots, \tau_t\}$ corresponding to the criteria in Θ, and a window w. τ_i is the base number of features to be selected for criterion θ_i. w is used to vary τ_i (thus the number of features) to be used by the wrapper approach.

Algorithm: EFS (F, Θ, T, w)

1. for each $\theta_i \in \Theta$
2. Rank all features in F based on criterion θ_i and let ξ_i denotes the ranked features
3. endfor
4. for $i = 1$ to t
5. $C_i \leftarrow \varnothing$
6. for $\tau = \tau_i - w$ to $\tau = \tau_i + w$
7. select first τ features ζ_i from ξ_i and add ζ_i to C_i in order
8. endfor
9. endfor
10. // $C_i = \{\zeta_1, \ldots, \zeta_{2w+1}\}$, where ζ_i is a set of features
11. OptCandFeatures $\leftarrow \varnothing$;
12. Repeat steps 13 – 18
13. $\Lambda \leftarrow \varnothing$
14. for $i = 1$ to t
15. select and remove the *first* feature set $\zeta_i \in C_i$ from C_i in order
16. $\Lambda \leftarrow \Lambda \cup \zeta_i$
17. endfor
18. add Λ to OptCandFeatures
19. // Λ is a set of features comprising of features in // feature sets $\zeta_i \in C_i$ in the same position $\forall\, i$
20. until $C_i = \varnothing\ \forall\, i$
21. for each $\Lambda \in$ OptCandFeatures
22. Λ.score $\leftarrow$ accuracy of 10-fold CV on training data on a chosen classifier (learning algorithm)
23. endfor
24. return $\arg\max_{\Lambda.score} \{\, \Lambda \mid \Lambda \in \text{OptCandFeatures}\}$

We now explain our EFS algorithm. Using a set of different feature selection measures, Θ, we rank all features in our feature pool, F, using the set of criteria (lines 1–3). This is similar to the filter approach. In lines 4–9, we generate feature sets C_i, $1 \le i \le t$ for each of the t criteria. Each set C_i contains feature subsets, and each subset ζ_i is the set of top τ features in ξ_i ranked based on criterion θ_i in lines 1–2. τ varies from $\tau_i - w$ to $\tau_i + w$ where τ_i is the threshold for criterion θ_i and w the window size. We vary τ and generate $2w + 1$ feature sets and add all such feature sets ζ_i to C_i (in lines 6–8) in order. We do so because it is difficult to know the optimal threshold τ_i for each criterion θ_i. It should be noted that "adding in order" ensures the ordering of feature sets ζ_i as shown in line 10, which will be later used to "select and remove in order" in line 15. In lines 11–20 we generate candidate feature sets using C_i and add each such

candidate feature set Λ to OptCandFeatures. Each candidate feature set Λ is a collection of top ranked features based on multiple criteria. It is generated by unioning the features in the *first* feature subset ζ_i, which is then removed from C_i for each criterion θ_i (lines 14-17). Each candidate feature set is added to OptCandFeatures in line 18. Since each C_i has $2w$+1 feature subsets ζ_i, there are a total of $2w$+1 candidate feature sets Λ in OptCandFeatures. Lines 21–23 assign an accuracy to each candidate feature set Λ ∈ OptCandFeatures by running 10-fold cross validation on the training data using a chosen classifier with the features in Λ. Finally, the optimal feature set Λ ∈ OptCandFeatures is returned in line 24.

An interesting question arising in the EFS algorithm is: How does one select the threshold τ_i for each criterion θ_i and the window size w? Intuitively, suppose that for criterion θ_i, the optimal subset of features is S_{opt_i} based on some optimal threshold τ_i. Then the final feature set is a collection of all features $f \in S_{opt\_i}\ \forall\ i$. However, finding such optimal feature set S_{opt_i} or optimal threshold τ_i is a difficult problem. To counter this, we use the window w to select various feature subsets close to the top τ_i features in ξ_i. Thus, the threshold values τ_i and window size w should be approximated by experiments. In our experiments, we used τ_i = top 1/20th of the features ranked in ξ_i for $\forall\ i$ and window size $w = |F|/100$, and got good results. Fortunately, as we will see in Section 6.2, these parameters are not sensitive at all, and any reasonably large size feature set seems to work equally well.

Finally, we are aware that there are some existing ensemble feature selection methods in the machine learning literature (Garganté et al., 2007; Tuv et al., 2009). However, they are very different from our approach. They mainly use ensemble classification methods to help choose good features rather than combining different feature selection criteria and integrating different feature selection approaches as in our method.

4.2 Feature Selection Criteria

The set of feature selection criteria $\Theta = \{\theta_1 \ldots \theta_t\}$ used in our work are those commonly used individual selection criteria in the filter approach.

Let $C = \{c_1, c_2, \ldots, c_m\}$ denotes the set of classes, and $F = \{f_1, f_2, \ldots, f_n\}$ the set of features. We list the criteria in Θ used in our work below.

Information Gain (IG): This is perhaps the most commonly used criterion, which is based on entropy. The scoring function for information gain of a feature f is given by:

$$IG(f) = -\sum_{i=1}^{m} P(c_i)\log P(c_i) + \sum_{f,\bar{f}} P(f)\sum_{i=1}^{m} P(c_i \mid f)\log P(c_i \mid f)$$

Mutual Information (MI): This metric is commonly used in statistical language modeling. The mutual information MI(f, c) between a class c and a feature f is defined as:

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

The scoring function generally used as the criterion is the max among all classes. MI(f) = max$_i$ {MI (f, c_i)} (which we use). The weighted average over all classes can also be applied as the scoring function.

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. We use a 2x2 contingency table of a feature f and a class c to introduce χ^2 test.

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

Table 4: Two-way contingency table of f and c

In the table, W denotes the number of documents in the corpus in which feature f and class c co-occur, X the number of documents in which f occurs without c, Y the number of documents in which c occurs without f, and Z the number of documents in which neither c nor f occurs. Thus, $N = W + X + Y + Z$ is the total number of documents in the corpus.

χ^2 test is defined as:

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

The scoring function using the χ^2 statistic is either the weighted average or max over all classes. In our experiments, we use the weighted average: $\chi^2(f) = \sum_{i=1}^{m} P(c_i)\chi^2(f,c_i)$

Cross Entropy (CE): This metric is similar to mutual information (Mladenic and Grobelnik,

1998):

$$CE(f) = P(f)\sum_{i=1}^{m} P(c_i \mid f)\log\frac{P(c_i \mid f)}{P(f)}$$

Weight of Evidence for Text (WET): This criterion is based on the average absolute weight of evidence (Mladenic and Grobelnik, 1998):

$$WET(f) = \sum_{i=1}^{m} P(c_i)P(f)\,|\log\frac{P(c_i \mid f)(1-P(c_i))}{P(c_i)(1-P(c_i \mid f))}|$$

5 Feature Value Assignments

After selecting features belonging to different classes, values are assigned differently to different classes of features. There are three common ways of feature value assignments: Boolean, TF (Term Frequency) and TF-IDF (product of term and inverted document frequency). For details of feature value assignments, interested readers are referred to (Joachims, 1997). While the Boolean scheme assigns a 1 to the feature value if the feature is present in the document and a 0 otherwise, the TF scheme assigns the relative frequency of the number of times that the feature occurs in the document. We did not use TF-IDF as it did not yield good results in our preliminary experiments.

The feature value assignment to different classes of features is done as follows: The value of F-measure was assigned based on its actual value. Stylistic features such words, and blog words were assigned values 1 or 0 in the Boolean scheme and the relative frequency in the TF scheme (we experimented with both schemes). Feature values for gender preferential features were also assigned in a similar way. Factor and word class features were assigned values according to the Boolean or TF scheme if any of the words belonging to the feature class exists (factor or word class appeared in that document). Each POS sequence pattern feature was assigned a value according to the Boolean (or TF) scheme based on the appearances of the pattern in the POS tagged document.

6 Experimental Results

This section evaluates the proposed techniques and sees how they affect the classification accuracy. We also compare with the existing state-of-the-art algorithms and systems. For algorithms, we compared with three representatives in (Argamon et al., 2007), (Schler et al., 2006) and (Yan and Yan, 2006). Since they do not have publicly available systems, we implemented them. Each of them just uses a subset of the features used in our system. Recall our system includes all their features and our own POS pattern based features. For systems, we compared with two public domain systems, *Gender Genie* (BookBlog, 2007) and *Gender Guesser* (Krawetz, 2006), which implemented variations of the algorithm in (Argamon et. al, 2003).

We used SVM classification, SVM regression, and Naïve Bayes (NB) as learning algorithms. Although SVM regression is not designed for classification, it can be applied based on the output of positive or negative values. It actually worked better than SVM classification for our data. For SVM classification and regression, we used SVMLight (Joachims, 1999), and for NB we used (Borgelt, 2003). In all our experiments, we used accuracy as the evaluation measure as the two classes (male and female) are roughly balanced (see the data description below), and both classes are equally important.

6.1 Blog Data Set

To keep the problem of gender classification of informal text as general as possible, we collected blog posts from many blog hosting sites and blog search engines, e.g., blogger.com, technorati.com, etc. The data set consists of 3100 blogs. Each blog is labeled with the gender of its author. The gender of the author was determined by visiting the profile of the author. Profile pictures or avatars associated with the profile were also helpful in confirming the gender especially when the gender information was not available explicitly. To ensure quality of the labels, one group of students collected the blogs and did the initial labeling, and the other group double-checked the labels by visiting the actual blog pages. Out of 3100 posts, 1588 (51.2%) were written by men and 1512 (48.8%) were written by women. The average post length is 250 words for men and 330 words for women.

6.2 Results

We used all features from different feature classes (Section 3) along with our POS patterns as our

pool of features. We used τ and w values stated in Section 4.1 and criteria mentioned in Section 4.2 for our EFS algorithm. EFS was compared with three commonly used feature selection methods on SVM classification (denoted by SVM), SVM regression (denoted by SVM_R) and the NB classifier. The results are shown in Table 5. All results were obtained through 10-fold cross validation.

Also, the total number of features selected by IG, MI, χ^2, and EFS were roughly the same. Thus, the improvement in accuracy brought forth by EFS was chiefly due to the combination of features selected (based on multi-criteria).

To measure the accuracy improvement of using our POS patterns over common POS n-grams, we also compared our results with those from POS n-grams (Koppel et al., 2002). The comparison results are given in Table 6. Table 6 also includes results to show the overall improvement in accuracy with our two new techniques. We tested our system without any feature selection and without using the POS sequence patterns as features.

The comparison results with existing algorithms and public domain systems using our real-life blog data set are tabulated in Table 7.

Also, to see whether feature selection helps and how many features are optimal, we varied τ and w of the EFS algorithm and plotted the accuracy vs. no. of features. These results are shown in Figure 1.

Feature Selection	Value Assignment	NB	SVM	SVM_R
IG	Boolean	71.32	76.61	78.32
IG	TF	66.01	72.84	74.13
MI	Boolean	72.01	78.62	79.48
MI	TF	70.86	73.14	74.58
χ^2	Boolean	72.90	80.71	81.52
χ^2	TF	71.84	73.57	75.24
EFS	Boolean	73.57	86.24	88.56
EFS	TF	72.82	82.05	83.53

Table 5: Accuracies of SVM, SVM_R and NB with different feature selection methods

Settings	NB	SVM	SVM_R
All features	63.01	68.84	70.03
All features, no POS patterns	60.73	65.17	66.17
POS 1,2,3-grams + EFS	71.24	82.71	83.86
POS Patterns + EFS	73.57	86.24	88.56

Table 6: Accuracies of POS n-grams and POS patterns with or without EFS (Boolean value assignment)

System	Accuracy (%)
Gender Genie	61.69
Gender Guesser	63.78
(Argamon et al., 2007)	77.86
(Schler et al., 2006)	79.63
(Yan and Yan, 2006)	68.75
Our method	88.56

Table 7: Accuracy comparison with other systems

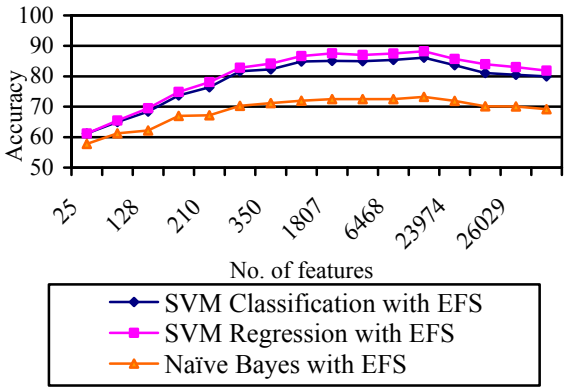

Figure 1: Accuracy vs. no. of features using EFS

6.3 Observations and Discussions

Based on the results given in the previous section, we make the following observations:

- SVM regression (SVM_R) performs the best (Table 5). SVM classification (SVM) also gives good accuracies. NB did not do so well.
- Table 5 also shows that our EFS feature selection method brings about 6-10% improvement in accuracy over the other feature selection methods based on SVM classification and SVM regression. The reason has been explained in the introduction section. Paired *t*-tests showed that all the improvements are statistically significant at the confidence level of 95%. For NB, the benefit is less (3%).
- Keeping all other parameters constant, Table 5 also shows that Boolean feature values yielded better results than the TF scheme across all classifiers and feature selection methods.
- Row 1 of Table 6 tells us that feature selection is very useful. Without feature selection (All features), SVM regression only achieves 70% accuracy, which is way inferior to the 88.56% accuracy obtained using EFS feature selection. Row 2 shows that without EFS and without POS sequence patterns, the results are even worse.

- Keeping all other parameters intact, Table 6 also demonstrated the effectiveness of our POS pattern features over POS n-grams. We have discussed the reason in Section 3.2 and 3.5.
- From Tables 5 and 6, we can infer that the overall accuracy improvement using EFS and all feature classes described in Section 3 is about 15% for SVM classification and regression and 10% for NB. Also, using POS sequence patterns with EFS brings about a 5% improvement over POS n-grams (Table 6). The improvement is more pronounced for SVM based methods than NB.
- Table 7 summarizes the accuracy improvement brought by our proposed techniques over the existing state-of-art systems. Our techniques have resulted in substantial (around 9%) accuracy improvement over the best of the existing systems. Note that (Argamon et al., 2007) used Logistic Regression with word classes and POS unigrams as features. (Schler et al., 2006) used Winnow classifier with function words, content word classes, and POS features. (Yan and Yan, 2006) used Naive Bayes with content words and blog-words as features. For all these systems, we used their features and ran their original classifiers and also the three classifiers in this paper and report the best results. For example, for (Argamon et al., 2007), we ran Logistic Regression and our three methods. SVM based methods always gave slightly better results. We could not run Winnow due to some technical issues. SVM and SVM_R gave comparable results to those given in their original papers. These results again show that our techniques are useful. All the gains are statistically significant at the confidence level of 95%.
- From Figure 1, we see that when the number of features selected is small (<100) the classification accuracy is lower than that obtained by using all features (no feature selection). However, the accuracy increases rapidly as the number of selected features increases. After obtaining the best case accuracy, it roughly maintains the accuracy over a long range. The accuracies then gradually decrease with the increase in the number of features. This trend is consistent with the prior findings in (Mladenic, 1998; Rogati and Yang, 2002; Forman 2003; Riloff et al., 2006; Houvardas and Stamatatos, 2006).

  It is important to note here that over a long range of 2000 to 20000 features, the accuracy is high and stable. This means that the thresholds of EFS are easy to set. As long as they are in the range, the accuracy will be good.

Finally, we would like to mention that (Herring and Paolillo, 06) has used genre relationships with gender classification. Their finding that subgenre "diary" contains more "female" and subgenre "filter" having more "male" stylistic features independent of the author gender, may obscure gender classification as there are many factors to be considered. Herring and Paolillo referred only words as features which are not as fine grained as our POS sequence patterns. We are also aware of other factors influencing gender classification like genre, age and ethnicity. However, much of such information is hard to obtain reliably in blogs. They definitely warren some future studies. Also, EFS being a useful method for feature selection in machine learning, it would be useful to perform further experiments to investigate how well it performs on a variety of classification datasets. This again will be an interesting future work.

7 Conclusions

This paper studied the problem of gender classification. Although there have been several existing papers studying the problem, the current accuracy is still far from ideal. In this work, we followed the supervised approach and proposed two novel techniques to improve the current state-of-the-art. In particular, we proposed a new class of features which are POS sequence patterns that are able to capture complex stylistic regularities of male and female authors. Since there are a large number features that have been considered, it is important to find a subset of features that have positive effects on the classification task. Here, we proposed an ensemble feature selection method which takes advantage of many different types of feature selection criteria in feature selection. Experimental results based on a real-life blog data set demonstrated the effectiveness of the proposed techniques. They help achieve significantly higher accuracy than the current state-of-the-art techniques and systems.

References

Agrawal, R. and Srikant, R. 1994. *Fast Algorithms for Mining Association Rules*. VLDB. pp. 487-499.

Argamon, S., Koppel, M., J Fine, AR Shimoni. 2003. *Gender, genre, and writing style in formal written texts*. Text-Interdisciplinary Journal, 2003.

Argamon, S., Koppel, M., Pennebaker, J. W., Schler, J. 2007. *Mining the Blogosphere: Age, Gender and the varieties of self-expression*, First Monday, 2007 - firstmonday.org

Baayen, H., H van Halteren, F Tweedie. 1996. *Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution*, Literary and Linguistic Computing, 11, 1996.

Blum, A. and Langley, P. 1997. *Selection of relevant features and examples in machine learning*. Artificial Intelligence, 97(1-2):245-271.

BookBlog, *Gender Genie*, Copyright 2003-2007, http://www.bookblog.net/gender/genie.html

Borgelt, C. 2003. *Bayes Classifier Induction*. http://www.borgelt.net/doc/bayes/bayes.html

Chung, C. K. and Pennebaker, J. W. 2007. *Revealing people's thinking in natural language: Using an automated meaning extraction method in open–ended self–descriptions*, J. of Research in Personality.

Corney, M., Vel, O., Anderson, A., Mohay, G. 2002. *Gender Preferential Text Mining of E-mail Discourse*. 18th annual Computer Security Applications Conference (ACSAC), 2002.

J. Dean and S. Ghemawat. 2004. *Mapreduce: Simplified data processing on large clusters*, Operating Systems Design and Implementation, 2004.

Forman, G., 2003. *An extensive empirical study of feature selection metrics for text classification.* JMLR, 3:1289 - 1306 , 2003.

Garganté, R. A., Marchiori, T. E., and Kowalczyk, S. R. W., 2007. *A Genetic Algorithm to Ensemble Feature Selection*. Masters Thesis. Vrije Universiteit, Amsterdam.

Gefen, D., D. W. Straub. 1997. *Gender differences in the perception and use of e-mail: An extension to the technology acceptance model*. MIS Quart. 21(4) 389–400.

Herring, S. C., & Paolillo, J. C. 2006. *Gender and genre variation in weblogs*, Journal of Sociolinguistics, 10 (4), 439-459.

Heylighen, F., and Dewaele, J. 2002. *Variation in the contextuality of language*: an empirical measure. Foundations of Science, 7, 293–340.

Houvardas, J. and Stamatatos, E. 2006. *N-gram Feature Selection for Authorship Identification*, Proc. of the 12th Int. Conf. on Artificial Intelligence: Methodology, Systems, Applications, pp. 77-86.

Joachims, T. 1999. *Making large-Scale SVM Learning Practical*. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

Joachims, T. 1997. *Text categorization with support vector machines*, Technical report, LS VIII Number 23, University of Dortmund, 1997

Kohavi, R. and John, G. 1997. *Wrappers for feature subset selection*. Artificial Intelligence, 97(1-2):273-324.

Koppel, M., Argamon, S., Shimoni, A. R.. 2002. *Automatically Categorizing Written Text by Author Gender*. Literary and Linguistic Computing.

Krawetz, N. 2006. *Gender Guesser*. Hacker Factor Solutions. http://www.hackerfactor.com/ GenderGuesser.html

Mladenic, D. 1998. *Feature subset selection in text learning*. In Proc. of ECML-98, pp. 95–100.

Mladenic, D. and Grobelnik, D.1998. *Feature selection for classification based on text hierarchy*. Proceedings of the Workshop on Learning from Text and the Web, 1998

Nowson, S., Oberlander J., Gill, A. J., 2005. *Gender, Genres, and Individual Differences*. In Proceedings of the 27th annual meeting of the Cognitive Science Society (p. 1666–1671). Stresa, Italy.

Riloff, E., Patwardhan, S., Wiebe, J.. 2006. *Feature Subsumption for opinion Analysis*. EMNLP,

Rogati, M. and Yang, Y.2002. *High performing and scalable feature selection for text classification*. In CIKM, pp. 659-661, 2002.

Schiffman, H. 2002. Bibliography of Gender and Language. http://ccat.sas.upenn.edu/~haroldfs/ popcult/bibliogs/gender/genbib.htm

Schler, J., Koppel, M., Argamon, S, and Pennebaker J. 2006. *Effects of age and gender on blogging*, In Proc. of the AAAI Spring Symposium Computational Approaches to Analyzing Weblogs.

Silva, J., Dias, F., Guillore, S., Lopes, G. 1999. *Using LocalMaxs Algortihm for the Extraction of Contiguous and Noncontiguous Multiword Lexical Units*. Springer Lecture Notes in AI 1695, 1999

Srikant, R. and Agrawal, R. 1996. *Mining sequential patterns: Generalizations and performance improvements*, In Proc. 5th Int. Conf. Extending Database Technology (EDBT'96), Avignon, France.

Tannen, D. (1990). *You just don't understand*, New York: Ballantine.

Tsuruoka, Y. and Tsujii, J. 2005. *Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data*, HLT/EMNLP 2005, pp. 467-474.

Tuv, E., Borisov, A., Runger, G., and Torkkola, K. 2009. *Feature selection with ensembles, artificial variables, and redundancy elimination.* JMLR, 10.

Yan, X., Yan, L. 2006. *Gender Classification of Weblog Authors*. Computational Approaches to Analyzing Weblogs, AAAI.