第 1 章 序論

1.1 研究の背景と目的

近年の Web の普及により，膨大な量の知識が Web 上に蓄えられるようになってきた．また，そこに蓄えられる知識の量は日々増加し続けている．そのような膨大な量の知識を蓄えている Web は，われわれの思考の基礎となる情報においても無視のできない存在になりつつある[29]．しかし，その Web における知識は一様な形式で記述されているわけではない．その多種多様な形式の中で最も多い形式が自然言語による記述である．人間は自然言語による記述を容易に理解することができるが，自然言語文をそのまま計算機によって解析することは困難である．

自然言語文においても，新聞などのある程度体裁の整った形式による記述であれば，形態素解析などを用いて計算機における解析が比較的容易にできる．しかし，Web における自然言語文は内容・用語・書式などが様々であり，計算機における解析は比較的容易とは言えない．

しかしそれでも，Web に記述される自然言語文の量は膨大であり，その膨大な量の知識を利用するために Web を対象とした研究が数多くなされてきた．Web における自然言語文からの新たな知識の獲得を目指す研究，Web におけるスパムサイトのフィルタリングの研究など，その内容は多岐にわたる．

Web において自然言語投稿文が記述される Web サイトは，情報の発信方法が手軽であるため，多くのユーザがそのような Web サイトを利用して自然言語投稿文を発信している．このようなサイトによって Web に蓄えられる知識の量は膨大であるため，現在このようなサイトに投稿される自然言語投稿文を対象とした研究が増えている．そのようなサイトの一つであるブログについては，既に海外において AAAI などで研究会が開かれているほどである．

今回の研究は，Web における自然言語投稿文を対象にテキスト分類を様々な方法で行うことによって，人間が Web の自然言語を扱っているサイトをより利用しやすく，より簡単に，そしてより価値の高い知識を獲得できるようにするための研究を行った．詳細は 1.2 節

で述べる.

1.2 本論文の構成

本論文では，Web における自然言語投稿文の学習に基づく分類により，人間がより容易に新たな知識を獲得できるようにすることを目指すため，以下に述べる 2 種類のアプローチを行う.

ひとつは Web において様々なユーザがその意見や考えを様々な内容・用語・書式などによって記述しているブログを対象とする分類である．日本におけるブログのユーザ数は総務省による「ブログ・SNS の現状分析および将来予測」[1]において，2005 年 3 月末で登録者数が述べ 335 万人，ユニークユーザ数が約 175 万人とされている．また，この資料による予測では 2006 年 3 月末では登録者数の延べ人数が予測では 621 万人であるが，2006 年 4 月に同じく総務省から発表された「ブログ及び SNS の登録者数」[2]では 868 万人と，予測を 40%ほど上回っており，2007 年 2 月の現時点ではすでに登録者数の延べ人数 1000 万人を突破していると思われる．海外においてもブログの利用者数は多く，ブログに関する研究会が何度も開かれているほど研究のホットトピックともなっている.

このように多くの人が利用し，その各々の意見を記述しているブログはまさに知識の宝庫であり，そこから知識を抽出しようという流れは世界的に広がっている．本論文においても，このブログを対象として，そのブログの書き手の性別を分類する研究を行った．ブログの書き手の性別の分類が可能になることによって，ブログに書き込まれた意見が男性女性どちら書き込んだ意見かを理解できるようになる．また，性別だけではなく，同じ手法で年齢別などの分類も装用の手法で行うことができる可能性が出てくる．これを本論文の第 2 章に記す.

第 3 章では，知識検索サイト，いわゆる QA サイトと呼ばれる Web サイトにおいて，不適切な投稿を分類する研究を行った．知識検索サイトとはあるユーザが質問を投稿し，別のユーザがその質問に答えるという形式のサイトである．代表的なものとして Yahoo!知恵袋[3]や教えて！goo[4]などが挙げられる. このようなサイトにはあるユーザが投稿した質問文と，それに対して他のユーザが投稿した回答という，人間が蓄えてきた Q&A の知識が自然言語文の形で大量に書き込まれている.

本論文では，知識検索サイトにおいて不適切な投稿文を分類する研究を行った．あらゆるユーザが質問を投稿することができる反面，想定されない投稿も行われる．その発見を計算機による解析で支援しようとするものである．Web における自然言語文は任意のユーザが書き込むために内容・用語・書式などが様々であり，コーパスが確実に正例と負例に分割できるとは限らないため，単純に正例と負例を用意して学習し分類するだけでは高い精度を得ることができない．このため，コーパスを精錬することによって分類精度をさらに高くすることを目指す. また，この手法は不適切な投稿文を分類するだけにとどまらず，

他の手法にも適用することができると思われる．正例・負例の一方のみが確実で他が不確実であるコーパスをもとに半教師つき学習を行うこのような研究は今までになかったため，本論文でこのような手法をもちいて自然言語投稿文を分類することを今後の様々な研究にとって有用なものとなると考えられる．

これらの研究を通じて Web における自然言語投稿文の学習に基づく分類，実験を行う．これらの研究について詳細を第 2 章，第 3 章で述べた後，第 4 章で本論文を締めくくる．

第 2 章 ブログへの投稿データを基とするブログ投稿者の性別自動推定

2.1 研究の背景と目的

日々更新されるインターネット上のブログには，ユーザ個人の意見がすばやく反映され，個人の意見が多く記述されるという特徴がある．ブログ記事から現在のトレンドを汲み取り，それを新商品開発や宣伝に活用する流れは，現在のマーケティング活動の中で大きな影響を及ぼしつつある．

しかし，ブログ投稿者の個人情報は一般的には公開されていない．ブログ投稿者が男性であるか女性であるかを判別できるようになれば，そのブログに書き込まれた意見が男性のものであるか女性のものであるか判別できるようになるため，マーケティング活動でも特に有用なものとなる．

また，計算機によって，文書から N-Gram を抽出しその表現の差を見ることで，書き手の性別により異なる単語を使用するという発見があった．[5]

本研究では, Doblog[30]におけるブログ記事とそのユーザに対して行った大規模なアンケート調査結果を対応付けて利用し，ブログ記事の文書から書き手の性別を判別するシステムを構築した．

以下，2.2 節ではブログに対するマイニングを行っている研究や様々なソースからその書き手の性別推定を行っている関連研究について述べる．2.3 節ではブログ記事の書き手の性別推定を行う本研究のシステム及び男性がよく用いる語・女性がよく用いる語の抽出について述べ，2.4 節で実験を行い，2.5 節でまとめる．

2.2 関連研究

自然言語で記述された文書を対象にその書き手の性別を推定する研究は，これまで数多く行われてきた．

1995 年に Support Vector Machine(SVM)による分類手法が提案されてから，SVM によ

りテキスト分類を行う研究は数多くなされてきた[11, 20]．なぜならば，それまで知られていたC4.5やk-NNといった既存の手法よりもよい分類精度を得られることが示されたためである[15].

[6]はBNCのテキストを用いて，著者の性別の推定を行っている．このテキストは，それぞれ34,000語あまりの長文の英語によって記述された文章である．素性としては，機能語と品詞，品詞列を使用しており，分類の精度としては80%ほどとなっている．また，この研究では，女性が代名詞を使用する頻度が高いことや，男性が数詞などの数を用いた表現を使用する頻度が高い，といった分析が行われている．

テキスト分類の手法を用いて自然言語文の筆者の性別推定を行う研究も行われてきた．[14]では言語の特徴を用いて性別の分類が行われた．[12]では言葉遣いを用いた分類が行われ，[17]では17世紀に記述された文書をもとに性別の分類が行われた．また，[13]では手紙のやり取りに用いられた文書を基に性別分類を行い，[19]では公式な文書では男女の性別による違いが見られないことが示されている．

[7]は，Eメールの送り主の性別判定を行っている．[6]でも行われているような機能語，品詞，品詞列を素性として用いる他にも，Eメールに使用されているHTMLタグや空行の数，文の平均長さなども素性に加えてSVMで分類を行っている．その結果，約7割の精度を得ている．

また近年，ブログを対象として計算機による解析や分類を行うような研究が盛んに行われている．ブログ2006年に行われたAAAIのシンポジウムでもこのようなセッションがあり，ブログのマイニングにより情報を得るといういくつかの発表があった．

[8]の研究では,’happy‘,’sad‘という2種類の感情に着目し，その感情が含まれていると思われる単語を抽出している．また，時間と’happy‘,’sad‘の感情が含まれている単語を利用して，ブログ投稿者がどの時間帯に最も’happy‘な感情になるか，もしくは’sad‘な時間帯であるかという推定も行っている．

また，[9]ではトラックバックなどのスパムをばらまくブログを，ブログからどのサイトにリンクしているかなどの素性を用いることによって同一性を確認し，85%ほどの精度をあげている．

AAAIのシンポジウムでも性別の制定に関する研究は発表された．[21]ではNaive Bayes分類器を用いてブログ投稿者の性別推定が行われた，素性としてはbag-of-wordsの単語やブログのデザイン(背景色，フォントの種類や色)，句読点や顔文字を用いた．

[18]ではブログ投稿者の性別や年齢を推定し，また各年代や性別でよく触れられている話題について示されている．例えば，スポーツや金融に関する単語は男性がよく用いており，睡眠，食事，家族や友達に関する単語は女性がよく用いている．また，10代はスポーツや友達に関する単語，20代では食事に関する単語，30代では金融や仕事，家族に関する単語がよく用いられていた．[16]ではN-Gramを素性に用いたSVM分類器を作成し，90%の分類精度を得た．

これらの研究は英語によって記述されたブログに関する研究であり，日本語のブログに関する研究はまだまだ少ないのが現状である．そのため，本研究では日本語のブログに対してブログ投稿者の性別推定を行うことを目的とする．

2.3 ブログ記事の性別推定

2.3.1 今回の提案手法の概要

今回の研究の性別推定は，4 つのフェーズに分けて行う．準備フェーズ，SVM 学習フェーズ，フィルタリングフェーズ，そして推定フェーズである．

- 準備フェーズ：ブログ記事から特徴ベクトルを生成する(2.3.2 節)
- SVM 学習フェーズでは，線形 SVM と非線形 SVM を，準備フェーズで特徴ベクトルを生成したブログ記事から作成する(2.3.3 節)
- フィルタリングフェーズでは，線形 SVM を作成することによって生成された学習器の各素性の重みを利用し，分類困難なブログ記事を事前に取り除く(2.3.4 節)
- 推定フェーズでは，SVM 学習フェーズで作成された非線形 SVM によってブログ投稿者の性別を推定する(2.3.5 節)

そして，2.3.6 節では，男性もしくは女性がよく使うと考えられる男性語，女性語の抽出を行うためのアプローチを紹介する．

2.3.2 特徴ベクトルの生成

今回の研究におけるコーパスは Doblog におけるブログ記事を用いた．はじめに，ブログ記事に対して形態素解析を行う．その際，単語の N-Gram(N=1～10)を取得した．N-Gram に用いる単語の品詞は名詞，動詞，形容詞である．これらの品詞に属する単語は，単語単体で意味を持つ．よって，これらの単語を用いることとする．

$$tfidf(t,b) = tf(t,b) \times idf(t) \tag{2.1}$$

$$tf(t,b) = \text{ブログ記事} b \text{ に単語} t \text{ が出てくる回数} \tag{2.2}$$

$$idf(t) = \log\left(\frac{N}{df(t)} + 1\right) \tag{2.3}$$

$$df(t) = \text{単語} t \text{ が出てくるブログ記事数} \tag{2.4}$$

$$N = \text{全ブログ記事数} \tag{2.5}$$

特徴ベクトルの各素性における値は $tfidf(t,b)$ (式(2.1))を用いる．また，素性として用いる

N-Gram は，$\sum_b tf(t,b) \geq 3$ となる，全出現回数が 3 回以上のものを用いることとする．また，全ての特徴ベクトルは正規化するものとする．

$$\mathbf{x}_b = \frac{\mathbf{x}'_b}{\| \mathbf{x}'_b \|}, \quad \mathbf{x}'_b = \begin{pmatrix} tfidf(t_1,b) \\ \vdots \\ tfidf(t_d,b) \end{pmatrix}, \quad d \text{ は全素性数} \tag{2.6}$$

すべてのブログ記事は，「Doblog の利用に関するアンケート調査」によって性別に関する回答を得たユーザのデータを用いている．今回の研究におけるコーパスのブログ記事は，すべてこの回答によって性別がラベル付けされたものとなっている．

2.3.3 線形 SVM/非線形 SVM 学習器の生成

2.3.2 節で用意された特徴ベクトルを用いて，線形 SVM と非線形 SVM をそれぞれ生成する．線形 SVM の式は以下のようになる．

$$prediction(\mathbf{x}) = \mathrm{sgn}[b + \mathbf{w}^{\mathrm{T}}\mathbf{x}] \tag{2.7}$$

非線形 SVM は以下のように，線形 SVM を(2.8)式で表されるカーネル関数と呼ばれる関数で拡張したものである．

$$prediction(\mathbf{x}) = \mathrm{sgn}[b + \sum_{i=1}^{m} w_i K(\mathbf{x}_i, \mathbf{x})] \tag{2.8}$$

$$K(\mathbf{x}_i, \mathbf{x}) = (1 + \mathbf{x}_i^{\mathrm{T}}\mathbf{x})^p \tag{2.9}$$

$\mathbf{x}_i$ はサポートベクタを表し，m はその数を表す．$K(\mathbf{x}_i, \mathbf{x})$ がカーネル関数と呼ばれる関数である．カーネル関数にはいくつかの種類があるが，今回は(2.9)式で表されるような 3 次の多項式カーネル($p = 3$)を用いることとした．なお，本研究では SVM のプログラムとして SVMlight[31]を用いている．

2.3.4 分類困難なブログ記事のフィルタリング

1) 分類困難なブログ記事

ブログ記事には確実に分類ができるほど素性が含まれていないものも含まれるために，すべてのブログ記事を分類器で分類するのは困難である．そのため，男性と女性の他に「分類困難なブログ記事」を定義し，男性と女性に分類する前にフィルタリングすることを考える．

分類困難なブログ記事として，以下の 2 種類が考えられる．

Confusable Post: 男性であると判断するための素性と女性であると判断するための素性が同程度入っているブログ記事である．これらの記事は SVM の分類境界面に近い

ということで判別できる．

Short Post: 分類するために必要な素性が記事中に少ないブログ記事である．これらの記事は特徴ベクトルにしたときに，値を持つ素性の数が非常に少ないということで判別できる．

これらのブログ記事をフィルタリングするために，分類における precision は上がるが，recall は下がる．

2）男性度，女性度

分類困難なブログ記事を判別するために，男性らしさと女性らしさをもっともらしく表した値として「男性度」と「女性度」を定義する．男性度と女性度は，素性の重みを利用した指標である．Brank らは素性選択の手法として線形 SVM を作成し，その際の素性の重みによって各素性が分類にどの程度効いているかを利用して素性選択を行った[10]．今回の研究では，この手法を利用して素性の重みを取得することとする．

重みの抽出を行う手順は以下のとおりである．

1. 2.3.3 節で学習された線形 SVM へ入力する特徴ベクトルとして，以下のものを用意する．

$$\mathbf{x} = (x_1, x_2, \ldots, x_d) \tag{2.10}$$

$$x_i = \begin{cases} 1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

2. ここで得られる $\mathbf{x}$ を式(2.6)に入力することによって，$b + w_t$ が値として返ってくる．ここで，線形 SVM を作成する際に $b = 0$ として学習器を作成するように指定することによって，計算を簡単にすることができる．

3. 先ほど得られた $b + w_t$ のうち，w_t を素性 t の重みとする．ここで，w_t が正である場合と負である場合で，その素性が男性によってよく用いられる語であるか，女性によってよく用いられる語であるかが分かる．今回の研究では，女性によるブログ記事を正例として，男性によるブログ記事を負例として扱った．これにより，w_t が正である場合は女性がよく用いる単語，負である場合は男性がよく用いる単語であることが考えられる．

各素性の重みは，$\mathbf{x}$ を入力することによって得られる．ここで各素性の重みを抽出できたことによって，各ブログ記事の男性度 s_m^b，および女性度 s_f^b を次のようにして計算することができる．

W_b = ブログ記事 b に含まれる素性の集合 (2.12)

$w(t)$ = 素性 t の重み (2.13)

$$s_f^b = \sum_{t \in T_f} w(t) \quad (2.14)$$

$$s_m^b = \sum_{t \in T_m} w(t) \quad (2.15)$$

$$T_f = \{\forall t \mid t \in W_b, w(t) > 0\} \quad (2.16)$$

$$T_m = \{\forall t \mid t \in W_b, w(t) < 0\} \quad (2.17)$$

3) 分類困難なブログ記事のフィルタリング

Confusable Post は男性度と女性度が近い記事だと仮定することができる．これを用いて，以下の式(2.18)によってフィルタリングすることができる． c_n はパラメータである．

$$|\log(s_f^b / s_m^b)| \geq c_n \quad (2.18)$$

Short Post は，男性度と女性度を用いて分類できる．なぜなら，Short Post はブログ記事に含まれる素性の数が少ないため，男性度や女性度も小さくなると考えられるからである．これにより，次の式(2.19)によってフィルタリングすることができる． c_s はパラメータである．

$$s_f^b \geq c_s \text{ or } s_m^b \geq c_n \quad (2.19)$$

これらの式(2.18)(2.19)で表されるフィルタによって，分類困難なブログ記事をフィルタリングすることができる．

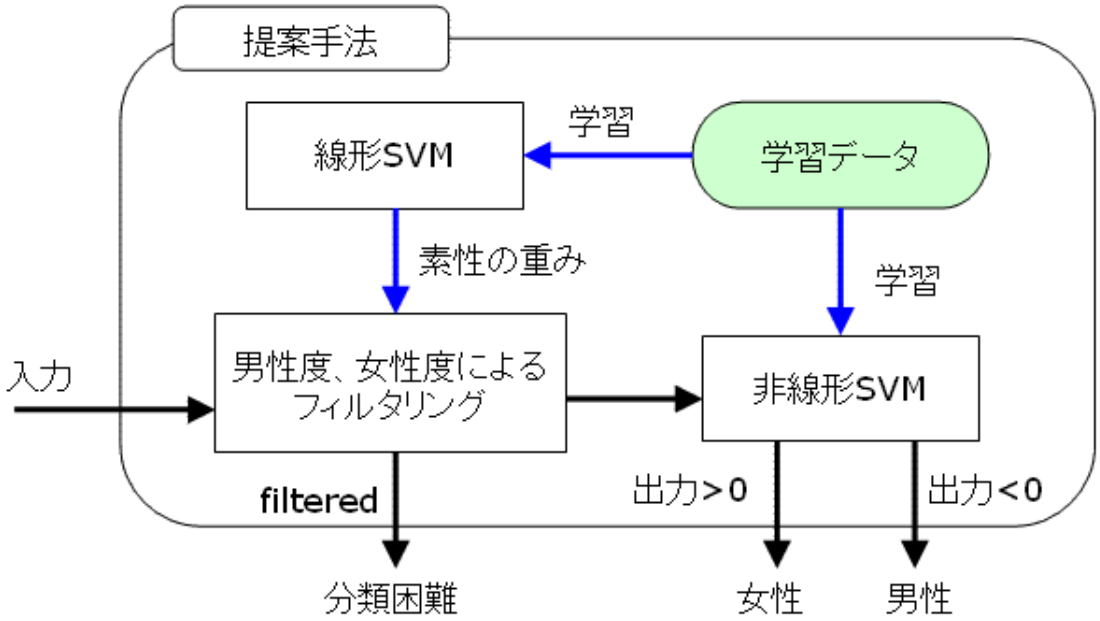

図 2.1 性別自動推定の流れ

2.3.5 ブログ投稿者の性別自動推定

以上より，性別自動推定の流れは以下のようになる．まず 2.3.2 節で得られた学習データの特徴ベクトルを用いて線形 SVM と非線形 SVM を学習する．そして，線形 SVM から各素性の重みを得，男性度，女性度によるフィルタリングの準備をする．

ブログ記事が入力されたら，まず男性度と女性度が計算される．これによってフィルタリングが行われ，式(2.18)(2.19)によってフィルタリングされた場合，そのブログ記事は分類困難なものであると出力される．そうでない場合は非線形 SVM による分類が行われ，出力が正であった場合は女性によって書かれたブログ記事，負であった場合は男性によって書かれたブログ記事であると出力される．全体の流れの図を図 2.1 に載せた．

2.3.6 男性もしくは女性によりよく用いられる単語

男性もしくは女性によりよく用いられる単語は，線形 SVM または *tfidf* によって抽出することができる．

前者は 2.3.4 節で用いた線形 SVM により得られた素性の重みを指標にする方法である．ここで得られた重みの大きさはどれだけどちらの性別に偏って出現しているかを表し，符号は男女どちらがそれを用いるかを表している．今回は正例を女性としたので，値が大きいほど女性がよく用いる単語であり，値が小さいほど男性がよく用いる単語であることが分かる．これにより，得られた素性の重みのリストをソートすることによって，男女が用いる単語のランキングを得ることができる．この手法により単語の重みの絶対値が大きか

った単語については 2.4.2 節にて述べる．

後者は男性，女性により書かれた記事の集合から *tfidf* を抽出し比較することによって男性，女性どちらがよく用いる単語であるかを判断する手法である．男性が書いたブログ記事の集合から得られた *tfidf* が大きければ男性のブログ記事によく用いられる単語であることが分かり，女性が書いたブログ記事の集合から得られた *tfidf* が大きければ女性のブログ記事によく用いられる単語であることが分かる．この手法によって得られた結果と考察は 2.4.6 節に載せる．

2.4 実験と結果

2.4.1 コーパス

今回の実験のコーパスとして，Doblog に投稿されたブログ記事と「Doblog の利用に関するアンケート調査」を用いた．どちらのデータも Doblog により提供された．Doblog は日本にて行われているサービスであり，提供されたブログ記事は日本語によって記述されている．これに含まれるブログ記事は全部で 241,251 記事であり，そのうち約 66%の 157,817 記事は男性によって書かれたものであり，83,434 記事は女性によって書かれたものであった．アンケート調査にはブログ投稿者の性別についての質問が含まれており，今回のコーパスに含まれるユーザはすべて男性か女性かどちらかの回答をしている．コーパスのユーザと対応付けることによって，コーパス中のブログ記事が男性によってかかれたものか女性によって書かれたものかが判別できるようになっている．

学習データとして，男女 1,000 記事ずつをコーパス中からランダムに取得した．また，それとは別に男女 10,000 記事ずつをテストデータとしてランダムに抽出した．

2.4.2 素性の選択

まず 3 種類の素性のセットによる学習データ/テストデータの分類精度の変化を比較することとした．

Noun: 素性の N-Gram を構成する品詞は名詞のみ

NAV: 素性として名詞(Noun)，形容詞(Adjective)，動詞(Verb)の 3 種類を用いたもの

NAV＋Tm: 名詞，動詞，形容詞に加えて文末表現を含むと考えられる助動詞，終助詞を素性として加えたもの

以上の素性のセットは，すべて出現回数が 3 回以上のもので構成されており，出現回数が 3 回未満となるものは取り除いてある．

これらの素性の異なる 3 種類の学習データ，テストデータによってフィルタリングを行

表 2.1 素性の種類と分類精度

素性の種類	Accuracy	素性の種類数
Noun	0.694	44261
NAV	0.693	67237
NAV+Tm	0.691	93685

表 2.2 重みの絶対値が大きい素性

素性	重み	素性	重み
私	3.93	僕	-2.78
笑	2.31	俺	-2.51
あたし	2.13	・	-1.84
わたし	1.86	0	-1.59
女	1.79	いうこと	-1.37
名前	1.78	東京	-1.35
食べる	1.41	曲	-1.34
コト	1.40	こちら	-1.33
今度	1.38	それ	-1.24
ワタシ	1.37	書く	-1.19

わずに分類を行った結果が表 2.1 である．Accuracy とはテストデータ全体の中で正しく分類された割合を表している．この実験によると 3 種類の素性のセットで，素性の種類数は違うが得られる分類精度に違いがないことが分かる．一番よい分類精度を得られるのは Noun であるが，今回の研究では動詞や形容詞を含むことにより得られる結果も重要であるため，NAV による素性のセットを用いる．

また，NAV を用いるときに，重みが大きい素性は表 2.2 のようになる．男女どちらも一人称代名詞に性別の特徴が出やすいということがわかる．関連研究[2]では女性が代名詞をよく使うという報告があったが，日本語のブログに関してあまり違いはないようである．また，[2]では男性が数字をよく用いるということも報告されていたが，表 2.2 では 0 という素性が上位に来ている．0 という素性が用いられる場合は，0 単独ではなく“100”や“200”など，数字の桁を表すものとしても用いられることが多いため，数字が用いられるときに 0 が入る確率は他の数字よりも高い．よって，日本語のブログにおいても男性が数字をよく用いるということがわかる．

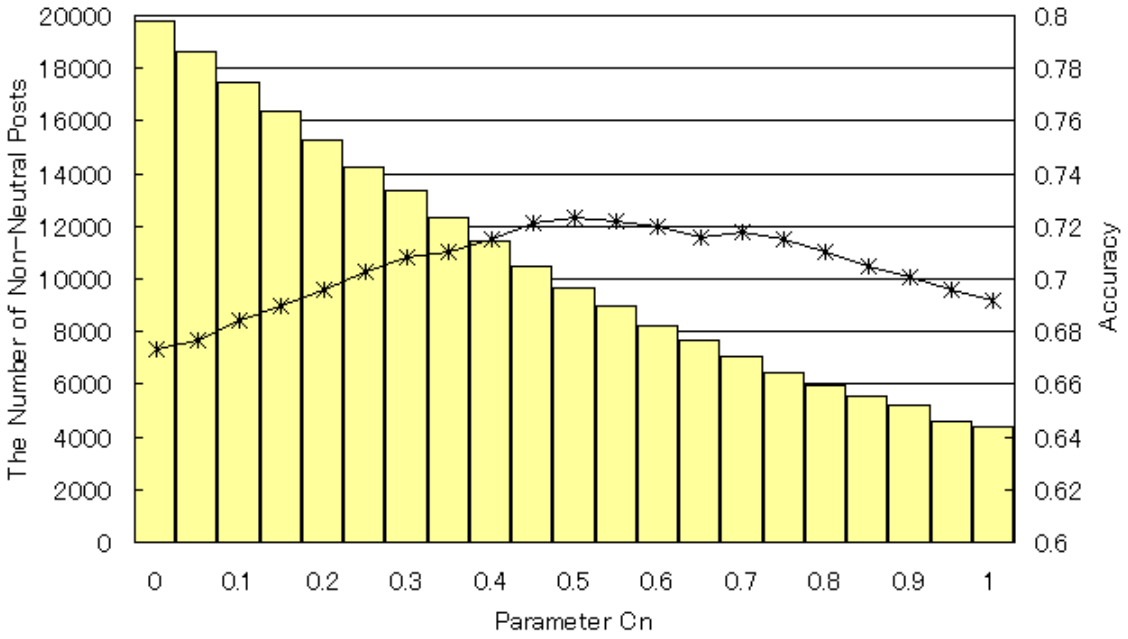

図 2.2 パラメータ c_n を変化させた時の分類精度の変化と
フィルタリングされずに残るブログ記事の数

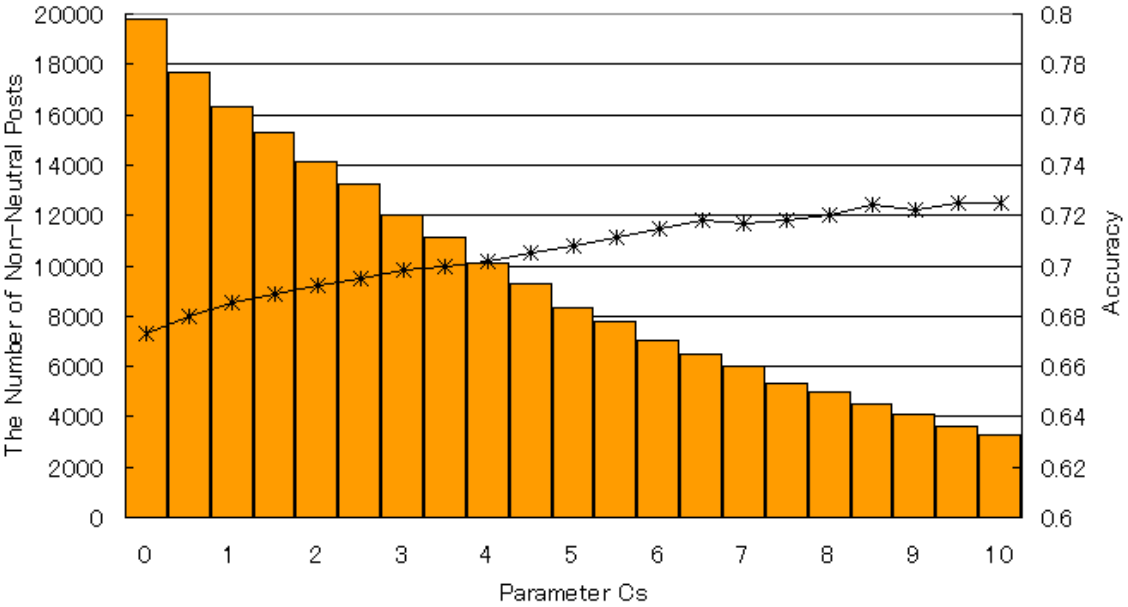

図 2.3 パラメータ c_s を変化させた時の分類精度の変化と
フィルタリングされずに残るブログ記事の数

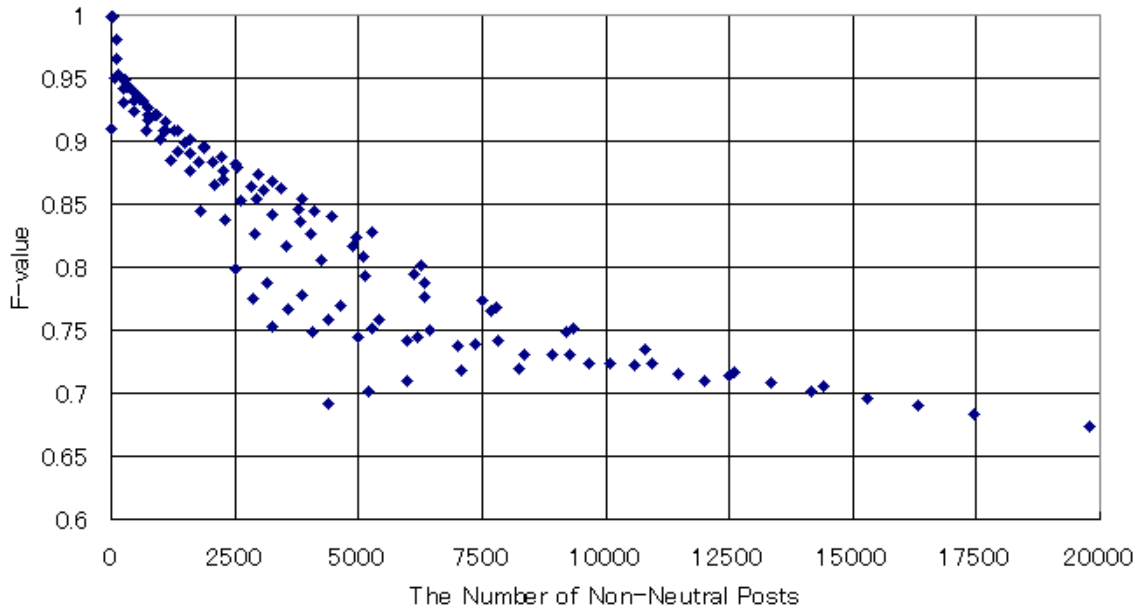

図 2.4 パラメータ c_n と c_s を共に変化させた時の分類精度と
フィルタリングされずに残るブログ記事の数の関係

2.4.3 フィルタリングのパラメータと分類精度

次に，フィルタリングを行い分類精度がどのようになるか調べる．式(2.18)(2.19)におけるパラメータ c_n， c_s を変化させ，分類困難なブログ記事を分類困難とすることによって残ったブログ記事の分類精度がどのようになるかを調べた．

まずは，パラメータ c_n のみを変化させて分類精度の変化を見た．この結果が図 2.2 である． c_n が増加するとフィルタリングされるブログ記事数が増加するため，フィルタリングされなかったブログ記事の数は減少している．分類精度に目を向けると， c_n が 0 から 0.5 までの間では c_n が増加すると分類精度も増加していることが分かる．しかし， c_n が 0.5 よりも大きくなると，分類精度は下がっている．

図 2.3 はパラメータ c_s のみを変化させたときの分類精度の変化である．こちらも c_s が増加するとフィルタリングされるブログ記事数が増加するため，フィルタリングされなかったブログ記事の数は減少する．分類精度は，図 2.2 のように途中から減少するということはないが， c_s の値が大きくなるにつれて飽和しつつあることが分かる．

最後に，パラメータ c_n と c_s の両方を変化させたときの分類精度と，フィルタリングされずに残るブログ記事との関係を調べる．パラメータ c_n と c_s で自由度が 2 つのため，同じ分類精度を与えるパラメータ (c_n, c_s) の組み合わせはいくつもある．同じ分類精度を与えるパラメータの組み合わせ同士を比較するには，その場合はフィルタリングされるブログ記事の数が少ないほうが優れていると考えられる．図 2.4 はこの実験を行ったときの結果である．これを見ると，パラメータ (c_n, c_s) の組み合わせでは，フィルタリングされるブログ記事数が増えることによって分類精度が高くなることがわかる．

2.4.4 パラメータの調整

本節では，分類精度を高めるためにパラメータ (c_n, c_s) の組み合わせを調節する．2.4.3 節のとおり，同じ分類精度となるパラメータ (c_n, c_s) の組み合わせは大量にあるが，それぞれフィルタリングされるブログ記事数が異なる．よって，同じ分類精度の中で最もフィルタリングされずに残るブログ記事数が最も多いものを調べることとする．

まず同じ分類精度となるパラメータ (c_n, c_s) の組み合わせをプロットし観察する．分類精度が F 値=0.8 となるパラメータの組み合わせをプロットした．この結果が図 2.5 である．これによると，パラメータ c_n と c_s は片方が小さいともう片方は大きくなければならないという，トレードオフの関係になっていることが分かる．

次に，パラメータ c_n と c_s のうち c_n と，フィルタリングされずに残るブログ記事の数を調べる．この結果が図 2.6 である．これによると，パラメータ c_n を増やしていくとあるところで最大となり，それより c_n を増やしてもフィルタリングされずに残るブログ記事の数は減少することが分かる．この原因を探る．図 2.2，図 2.3 よりパラメータ c_n， c_s どちらが単体で大きい場合でもフィルタリングされるブログ記事数が増える．それを今節に当ては

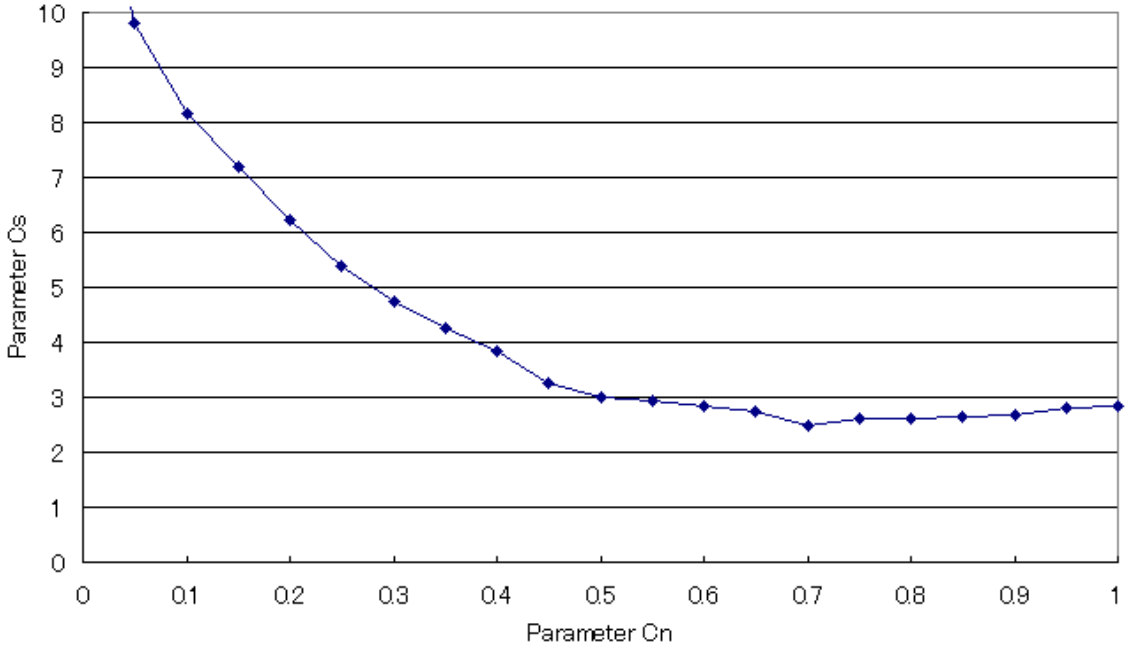

図 2.5　F 値=0.8 となる時のパラメータ(c_n, c_s)の組み合わせを
プロットした時に描く曲線

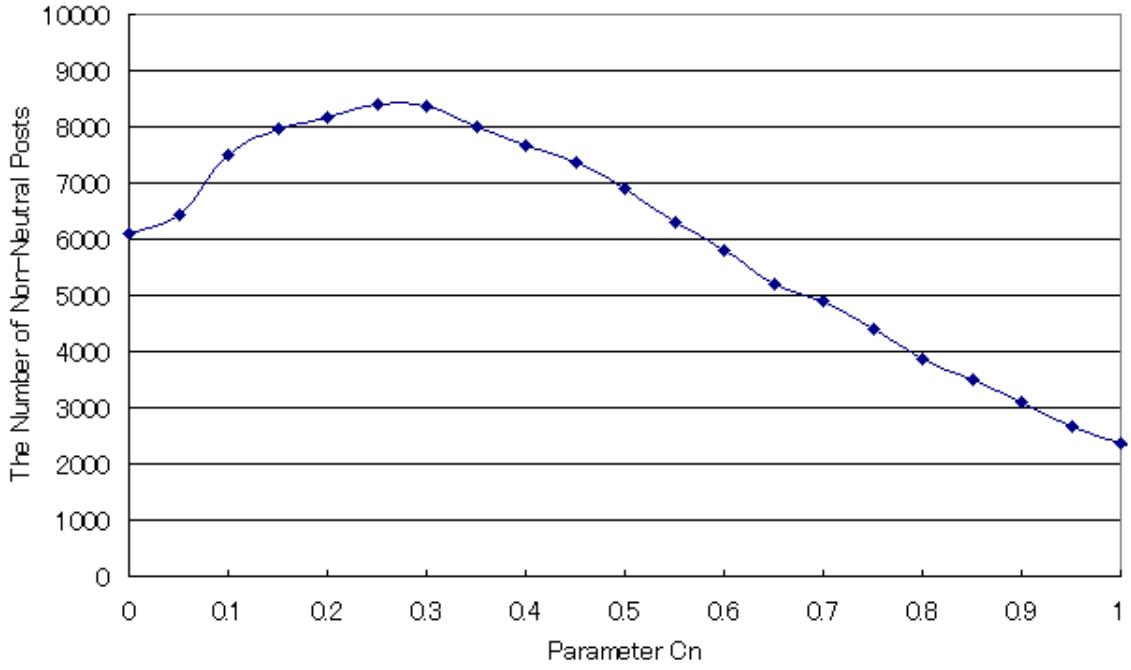

図 2.6 F 値=0.8 となるパラメータ(c_n, c_s)の組み合わせのうちc_nと，
フィルタリングされずに残るブログ記事の数

めると，パラメータ c_n が大きい場合はそれによってフィルタリングされるブログ記事数が増えるためにフィルタリングされるブログ記事数が増える．逆にパラメータ c_n が小さい場合には，図 2.5 の結果によりパラメータ c_s が大きくなるために，その影響でフィルタリングされるブログ記事数が増加すると考えられる．

以上により，フィルタリングされずに残るブログ記事数の最大値を与えるパラメータ (c_n, c_s) の組み合わせが存在することが分かった. 今回の実験を詳しくやった結果, F 値=0.8 においてフィルタリングされずに残るブログ記事数の最大値を与えるパラメータ (c_n, c_s) の組み合わせは (c_n, c_s)=(0.24, 5.55)であった．

表 2.3 種々のデータセット間の *tfidf* リストのスピアマン順位相関係数

	B_m	B_f	B_m^*	B_f^*
B_m		0.742	0.853	0.622
B_f	0.742		0.584	0.875
B_m^*	0.853	0.584		0.405
B_f^*	0.622	0.875	0.405	

2.4.5 *tfidf* の抽出による検証

2.4.4 節で得られた (c_n, c_s)=(0.24, 5.55)という組み合わせが得られた．これによって得られたブログ記事はフィルタリングされ, その後非線形 SVM で分類されることになる. では, これによって得られた結果は，確かに元の文書集合に近いかという実験を行った．

B_m，B_f はそれぞれ男性, 女性によって書かれたテストデータ 10,000 記事ずつ, B_m^*，B_f^* はフィルタリングを行い非線形 SVM で男性, 女性と分類されたそれぞれ 2,973 記事, 4,135 記事の集合である．これらの記事集合から素性の *tfidf* を抽出し順位付けを行い，スピアマンの順位相関係数で比較を行う．2 つのテストデータ間の記事集合の類似性が高い場合，この相関係数も高くなるはずである．

結果は表 2.4 のとおりである．これを見ると B_m，B_m^* 間，B_f，B_f^* 間の順位相関係数は他に比べて高いことが分かる．これにより，このシステムによって男性，女性だと分類さ

れた記事集合は，もとの男性，女性テストデータの記事集合との類似性が高くなっていることが分かる．

2.4.6 ユーザに着目した性別自動推定

今までのタスクにおいて，ブログ記事単体に着目して分類を行ってきた．しかしブログ単体では文書量が少なく，分類困難となるブログ記事が多いなどの困難がある．そこでブログ記事単体ではなく，ユーザごとにまとめて分類を行うことを提案する．Doblog からいただいたコーパスには 241,251 記事のブログ記事が含まれているが，それらは 752 人のユーザによって投稿されたブログ記事である．その中で男性であるユーザ数は 480 人，女性であるユーザ数は 272 人である．1 人あたり平均して約 320 記事のブログ記事を書いている．今節では同じユーザによって書かれたブログ記事をひとまとめにして分類する手法について提案し，実験を行う．

ユーザごとに N 件のブログ記事をひとつにまとめて一つのブログ記事とみなし，図 2.1 の流れでフィルタリングせずに分類した際の分類精度を見る．この結果は図 2.7 である．およそ 30 件のブログ記事をまとめると分類精度の上昇が飽和していると見られる．

次に，ブログ記事単体のときと同じように，フィルタリングによって分類精度を上げることを考える．先ほどの実験結果より，1 ユーザに対して 30 件ほどのブログ記事をまとめれば十分な精度が出ると考えられる．そうしてまとめた 1 ユーザの記事集合 u に対して，式(2.12)～(2.17) を適用することによって，各ブログ記事と同様にユーザの男性度 s_m^u，および女性度 s_f^u を求めることができる．

このままでもよいのだが，男性度と女性度の値で，女性度のほうが小さく出てしまうという問題が生じることが分かった．そこで，式(2.12)～(2.17)を拡張して以下の式を定義することとした．

$$W_u = \text{ユーザ}\, u \,\text{の記事集合に含まれる素性の集合} \tag{2.20}$$

$$w(t) = \text{素性}\, t \,\text{の重み} \tag{2.21}$$

$$s_f^u = \sum_{t \in T_f} w(t) \times tfidf(t,u) \tag{2.22}$$

$$s_m^u = \sum_{t \in T_m} w(t) \times tfidf(t,u) \tag{2.23}$$

$$T_f = \{\forall t \mid t \in W_u, w(t) > 0\} \tag{2.24}$$

$$T_m = \{\forall t \mid t \in W_u, w(t) < 0\} \tag{2.25}$$

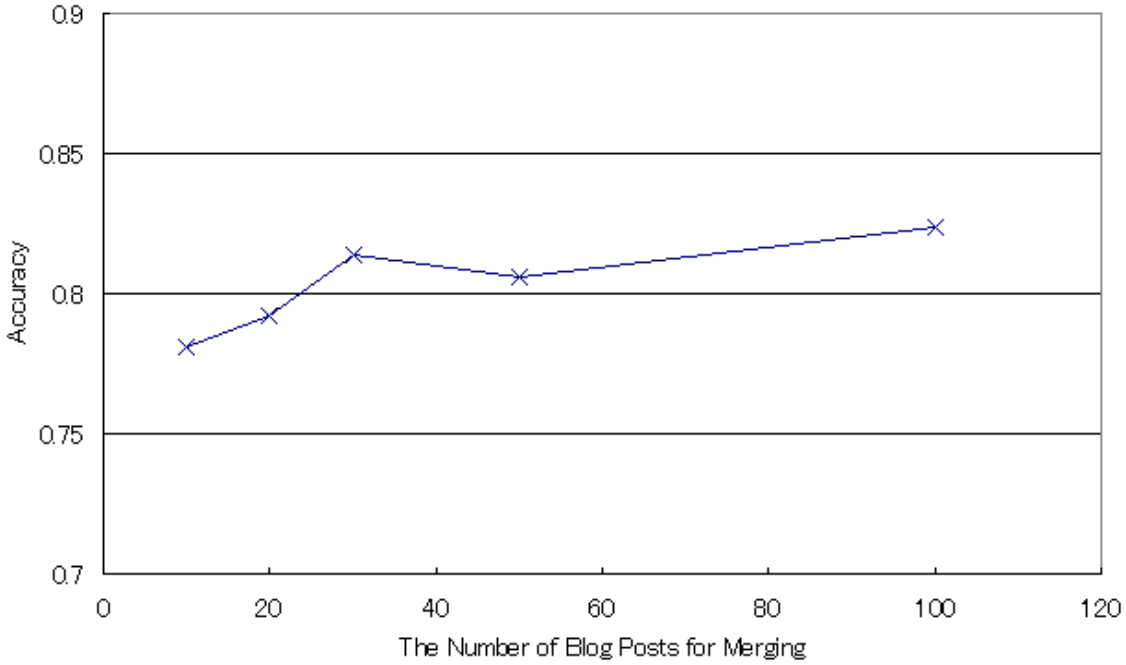

図 2.7 ユーザごとにひとまとめにしたブログ記事の数と分類精度の変化

表 2.4　フィルタリングの種類と分類可能なユーザ数

フィルタ	パラメータ	Accuracy	Male/Female/Total	Coverage
なし	なし	0.811	480/ 272/ 752	1.000
DS + small	c_s = 150	0.854	281/ 186/ 467	0.621
DS + frac	c_n = 0.07	0.850	398/ 177/ 575	0.764
DS + margin	c_n = 7	0.845	409/ 197/ 606	0.801
TS + small	c_s = 3.5	0.844	390/ 240/ 630	0.838
TS + frac	c_n = 0.04	0.854	438/ 247/ 685	0.911
TS + margin	c_n = 0.12	0.851	441/ 251/ 692	0.920
DS + small	c_s = 200	0.894	136/ 101/ 237	0.315
DS + frac	c_n = 0.15	0.898	227/ 53/ 280	0.372
DS + margin	c_n = 20	0.895	309/ 100/ 409	0.544
TS + small	c_s = 4.2	0.899	226/ 269/ 395	0.525
TS + frac	c_n = 0.08	0.899	401/ 231/ 632	0.840
TS + margin	c_n = 0.3	0.900	396/ 232/ 628	0.835

ここで混乱を避けるために，式(2.12)～(2.17)を利用して算出された男性度，女性度を‘DS-Score’，式(2.20)～(2.25)を利用して算出された男性度，女性度を‘TS-Score’と名づける．TS-Score とは，*tfidf* を利用して算出されたというところから名づけた．

また，フィルタリングに関する式(2.18)(2.19)についても，以下のように変更する．

$$s_f^u \geq c_s \text{ or } s_m^u \geq c_n \tag{2.25}$$

$$|\log(s_f^u / s_m^u)| \geq c_n \tag{2.26}$$

式(2.26)については，対数関数を用いることによる影響をみるために，対数関数を用いない以下の式によるフィルタリングも行った．

$$|s_f^u - s_m^u| \geq c_n \tag{2.27}$$

ここで，式(2.25)によるフィルタリングを‘small-filtering’，式(2.26)によるフィルタリングを‘frac-filtering’，式(2.27)によるフィルタリングを‘margin-filtering’と名づける．例えば，男性度，女性度の求め方を‘DS-Score’，フィルタリングの式を式(2.25)で行った場合，‘DS + small’によるフィルタリングを行った，ということとする．

以上によって実験を行った．まずフィルタリングがない場合，1 ユーザから 30 のブログ記事をまとめて 1 つの記事とみなしてフィルタリングを行わずに分類した場合, 図 2.7 より Accuracy は 0.811 となる．

そして，‘DS-Score’，‘TS-Score’および式(2.25)～(2.27)によるフィルタリングをすることにより，すべてのフィルタリングの組み合わせにおいて Accuracy が 0.85，および 0.9 付近になるように調節することができた．この結果が表 2.4 である．フィルタリングされずに男性か女性か分類することができる割合が Coverage となっており，Accuracy が 0.85 よりも 0.9 としたときに Coverage は下がっている．

フィルタリング手法間の Coverage の差を見てみると，以下のことが分かる

- ‘DS-Score’と比べて‘TS-Score’のほうが Coverage は高い
- ‘small-filtering’よりも‘frac-filtering’，‘margin-filtering’のほうが Coverage は高い

この実験により，全体の 92%のブログ投稿者を 85%の精度で，及び 84%のブログ投稿者を 90%の精度で分類することができた．

2.4.7 男性または女性によりよく用いられる単語

2.3.6 節で述べ 2.4.5 節でも利用した，男性が書いたブログ記事の集合及び女性が書いたブログ記事の集合から得られた単語の *tfidf* を比較し，その単語が男性，女性どちらに特徴的に出てくるかを見ることとする．これらの *tfidf* 値の一覧は表 2.5，表 2.6a および表 2.6b

表 2.5 男性によるブログ記事，女性によるブログ記事から得られる
名詞のキーワードによる *tfidf*

Keyword	女性文書	男性文書
彼氏	255	6
彼女	143	211
夫婦	292	280
両親	10	6
母親	15	67
父親	15	9
親子	77	26
親戚	117	2
親族	0	122
親友	78	1
友達	208	165
友達 関係	1065	0
いい 友達	107	0
ピンク色	1	124
ピンク	51	2
金色	94	0
黄色	3	21
黄色い	51	6
野球	538	330
プロ 野球	73	341
野球 選手	383	2433
サッカー	275	424
サッカー 日本 代表	1	290
高校 サッカー	0	1684
テニス	62	141
バレーボール	55	0
バスケット	0	128
駅伝	78	0
競馬	51	307

表 2.6a 男性によるブログ記事，女性によるブログ記事から得られる
動詞のキーワードによる *tfidf*

Keyword	女性文書	男性文書
歩く	122	184
走る	71	0
考える	191	244
出かける	232	76
買う	530	942
売る	96	160
食べる	795	1084
食べる られる	62	205
食べる れる	123	320
食べる てる	132	386
食べる ちゃう	243	1
食べる させる	107	4
食べる よう	199	1
食べる 始める	1652	4354
食べる 終わる	5321	2351
食べる 続ける	75	5119
食べる まくる	59	0
食べる すぎる	176	1
食べる 過ぎる	2435	9189
食べる 残す	0	3136
良い 食べる	1713	0
美味しい 食べる	718	1701
美味しい 食べる られる	470	0

表 2.6b 男性によるブログ記事，女性によるブログ記事から得られる
動詞のキーワードによる *tfidf*

Keyword	女性文書	男性文書
毎日 食べる	108	0
全部 食べる	1713	0
もの 食べる	255	533
ご飯 食べる	7130	5142
ごはん たべる	90	0
夕飯 食べる	465	1
ケーキ 食べる	872	1
匹 食べる	11	0
の 食べる	0	212
食べる 方	83	339
食べる もの	73	130
食べる の	39	4
食べる 事	102	0
食べる こと	5	278
食べる とき	598	0
食べる てる とき	166	0
食べる 時間	79	0

に載せてある．

表 2.5 は名詞句に関する単語に関する *tfidf* 値の一覧である．まず，スポーツに関する単語についてみてみると，大部分の単語については男性文書から得られる *tfidf* 値のほうが大きいことが分かる．[18]によると英語のブログにおいてもスポーツに関する単語は男性のほうがよく用いていた．ただし，一部の「バレーボール」や「駅伝」といった単語については女性文書から得られる *tfidf* 値のほうが大きい．

また，色を表す単語を見てみた．「ピンク」「ピンク色」と「黄色」「黄色い」はそれぞれ同じ内容を表す単語であると考えられるが，*tfidf* 値は男性文書のほうが高い，女性文書のほうが高い，とそれぞればらばらになっており男性と女性で同じ色を表すときに異なる表現を用いているという結果が出た．

人間関係を表す単語については，「親友」などの友達関係を表す単語や「彼氏」という単語が女性からよく抽出され，「彼女」などの単語は男性からよく抽出された．「親族」「親戚」は色を表す単語のときと同じように，同じ内容をさしていると考えられるがそれぞれ男性文書，女性文書の一方のみからよく抽出された．

表 2.6a，表 2.6b は動詞についてみてみたときの *tfidf* 値の一覧である．表 2.6a の上部はいくつかの動詞である単語を見てみたときの *tfidf* 値である．男性も女性とることができる行動は大体同じであるため，いずれの単語についても *tfidf* 値が同じであると予想される．しかし表 2.6a をみると，「走る」や「出かける」などいくつかの単語で女性文書からの *tfidf* 値が大きくなったり，その逆があったりすることが分かる．

表 2.6a の下部と表 2.6b は，「食べる」という動詞が含まれた N-Gram をすべてとってきて，「食べる」にさまざまな単語がついた時，男性文書から得られる *tfidf* 値と女性文書から得られる *tfidf* 値のどちらが大きくなるかを調べたものである．これを見ると，同じ「食べる」ことを主とした動作でもあるのに関わらず，前後につく名詞や動詞によって男女の *tfidf* 値に差がつくことが分かる．

2.5 議論

本論文では Web において自然言語文で記述された投稿文としてブログのデータを解析し，そのブログが男性女性どちらの性別で書かれたものかの推定を行った．ブログの特徴である，1 つのブログにいくつかの記事がまとまっている形式を利用し，高い精度でブログ投稿者の性別推定を行うことができた．

また，得られたデータを利用し，男性または女性がよく使うと思われる単語の抽出方法について紹介した．これは男性，女性がよく用いる単語に限らず，年代別やの単語の抽出，およびそれらをクラスタリングすることによってあるドメインの人々がどのような話題に興味を持っている傾向があるかを判断することができると考えられる．本論文では触れな

いが，そのような研究が出てくると面白いだろう．

ブログ記事単体の分類については，ブログのみに用いるものとしての有用性は低いかもしれない．しかし，ブログに限らない Web における自然言語投稿文，例えば一般の掲示板などの 1 投稿は文書の長さがブログ程度になるものもあり，ブログ記事単体の分類の手法をそのまま適用することもできると考えられる．掲示板に適用できるならば，その投稿の書き手の性別を判断することができるようになるため，より広範囲の投稿文に対して性別のタグ付けが可能になるだろう．また，名前記入式の掲示板であれば，そのユーザの過去の発言を今回の手法と同じように一つにまとめ，分類することで精度の高い分類を行うことができるようになる．

第 3 章 知識検索サイトへの不適切投稿の自動推定

3.1 はじめに

近年，Web の急速な普及によって膨大な量の知識が Web 上に蓄積されるようになってきた．このような知識を利用したサイトの一つとして，知識検索サイトがあり，ユーザは質問や回答を投稿することができる．しかし，全てのユーザが想定された適切な行動をとるとは限らないため，不適切な投稿(スパム)の排除は欠かせない．

人手でラベル付けされている投稿についても，不適切な投稿であるかどうかを完全に判断することは難しい．知識検索サイトはその性質上，完全に不適切だと分かる投稿文について削除しており，削除されていない投稿文が完全に不適切ではないと言い切れないため，それらの文書をコーパスとして用いるのはコーパスとしてはやや不完全であり，これが分類精度の向上を阻害している可能性がある．

そこで，本研究では現在人手で行っている不適切な投稿の削除を機械学習による不適切投稿の発見を半自動化するために，教師つき負例と未知データからなる半教師つき学習コーパスからの SVM (Support Vector Machine)学習器の作成に取り組んだ．

以下，3.2 節では知識検索サイトについて述べる．3.3 節で本研究の手法について述べ，3.4 節で実験とその考察を行い，3.5 節で関連研究について述べ，まとめる．

3.2 知識検索サイト

3.2.1 知識検索サイトの特徴

知識検索サイトとは，あるユーザが投稿した質問に他のユーザが回答を投稿するサイトであり，お互いに知恵や知識を教え合うことを目的としている．過去に投稿された質問や回答は記録されており，後から検索することによって直接その質問と回答には関係ないユーザも情報を得ることもできる．このようなサイトとして，Yahoo!知恵袋[3]や教えて！goo[4]などがある．

3.2.2 禁止行為

Yahoo! 知恵袋には，利用する際のガイドラインがある．その中に禁止行為というものが示されており，それらに抵触する投稿は削除されるとされている［25］．このガイドライン中には，以下のような投稿が禁止行為として示されている．

- いやがらせ，悪口，脅し，あるいは有害な内容の掲載など，他人を攻撃したり，傷つけたりする目的で利用すること
- わいせつな内容や不愉快なデータを公開すること
- 商業目的や広告目的で利用すること
- 質問と関係のないことを書き込むこと
- Yahoo! JAPAN が予定していない目的で本サービスを利用すること
- 著作権者の許可を受けずに著作物を公開するなど，第三者の知的財産権を侵害したり，侵害を助長すること
- プライバシー侵害の恐れがある事実やデータを公開すること
- その他，Yahoo! JAPAN が不適切だと判断する行為

3.3 提案手法

3.3.1 実験用コーパスの作成

本研究で用いるコーパスについて考察する．削除された質問文書は不適切であることがわかるため，これを負例として用いる．正例は，削除されていない質問文書を用いる．

本研究では，不適切であることが判明している文書と，適切・不適切が混ざった未知の文書があったときに，コーパスを精錬することによって如何にして精度の高いコーパスを得るか，という問題に取り組む．そのため，コーパスに含まれる正例と負例を混ぜ，教師つき負例と未知データからなる半教師つき学習コーパスを作成する．

3.3.2 コーパスの精錬手法

1) スパム分離型の精錬手法

まず，未知コーパスから不適切な文書(スパム)を取り除いていき，コーパスを精錬していく手法について述べる．そして，以下のような手順で精錬を行う．

1. コーパスから，教師つき負例データセット D_0，２組の未知データセットそれぞれ P_0^A と P_0^B を抽出する．
2. n は $n \geq 0$ の整数とする．D_{2n} と P_{2n}^A から文書を１：１にランダムに抽出して SVM 学習器を作成し，P_{2n}^B を分類する．これによってスパムだと判定されたもののうちある条件 Q を満たすものを S_{2n}^B とし，残りを N_{2n}^B とする．
3. $D_{2n+1} = D_{2n}$ ， $P_{2n+1}^A = P_{2n}^A$， $P_{2n+1}^B = N_{2n}^B$ とする．

4. D_{2n+1} と P_{2n+1}^{B} から文書を１：１にランダムに抽出して SVM 学習器を作成し，P_{2n+1}^{A} を分類する．これによってスパムだと判定されたもののうちある条件 Q を満たすものを S_{2n+1}^{A} とし，残りを N_{2n+1}^{A} とする．
5. $D_{2n+2} = D_{2n+1}$ ， $P_{2n+2}^{A} = N_{2n+1}^{A}$， $P_{2n+2}^{B} = P_{2n+1}^{B}$ とする．
6.2.～5.を， S として取り除かれる投稿文が収まるまで繰り返す．

この時, 取り除かれた文の集合を $S = S_0^B + S_1^A + S_2^B + S_3^A + \cdots$ と定義する. ある条件 Q としては，SVM の出力がある一定値を越える，スコアの大きい M 件を取り除く，などの方法によって決めることができる．また，任意の n において $D_n = D_0$ となる．

2）負例追加型の精錬手法

この手法は，1)のスパム分離型の手法と似ている．これと異なる点は，スパムだと判定されて取り除かれていた S_{2n}^{B} 等を，単純に取り除く代わりに D_0 に追加していくということである．1)のアルゴリズムに以下の改良点を加えたものが負例追加型の精錬手法となる．

3. $D_{2n+1} = D_{2n} + S_{2n}^{B}$ ， $P_{2n+1}^{A} = P_{2n}^{A}$， $P_{2n+1}^{B} = N_{2n}^{B}$ とする．
5. $D_{2n+2} = D_{2n+1} + S_{2n}^{A}$ ， $P_{2n+2}^{A} = N_{2n+1}^{A}$， $P_{2n+2}^{B} = P_{2n+1}^{B}$ とする．

任意の n において $D_n = D_0 + S$ となる．スパム分離型，負例追加型の精錬手法の概略図については図 2.1 に載せた．

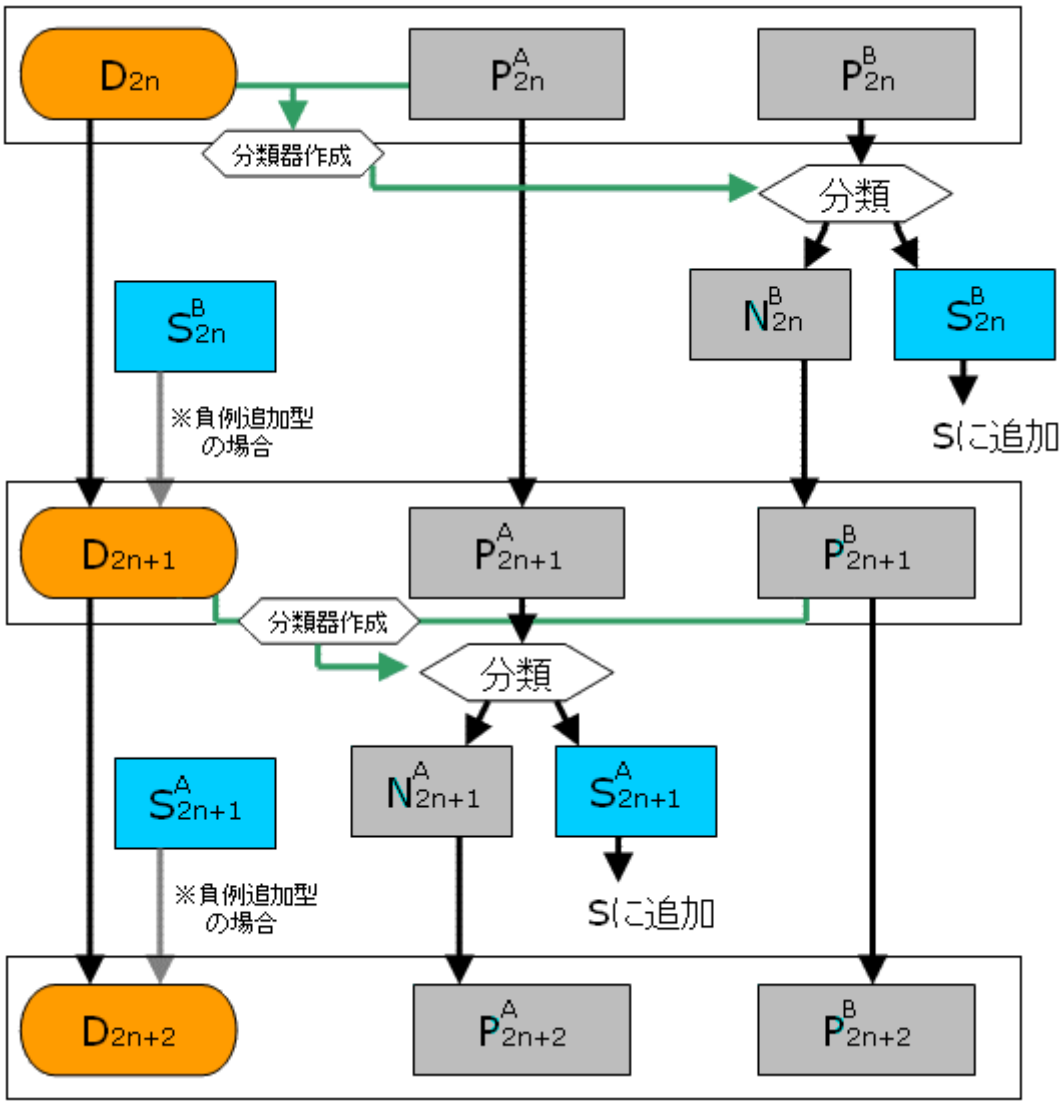

図 3.1 スパム分離型及び負例追加型のコーパス精錬方法

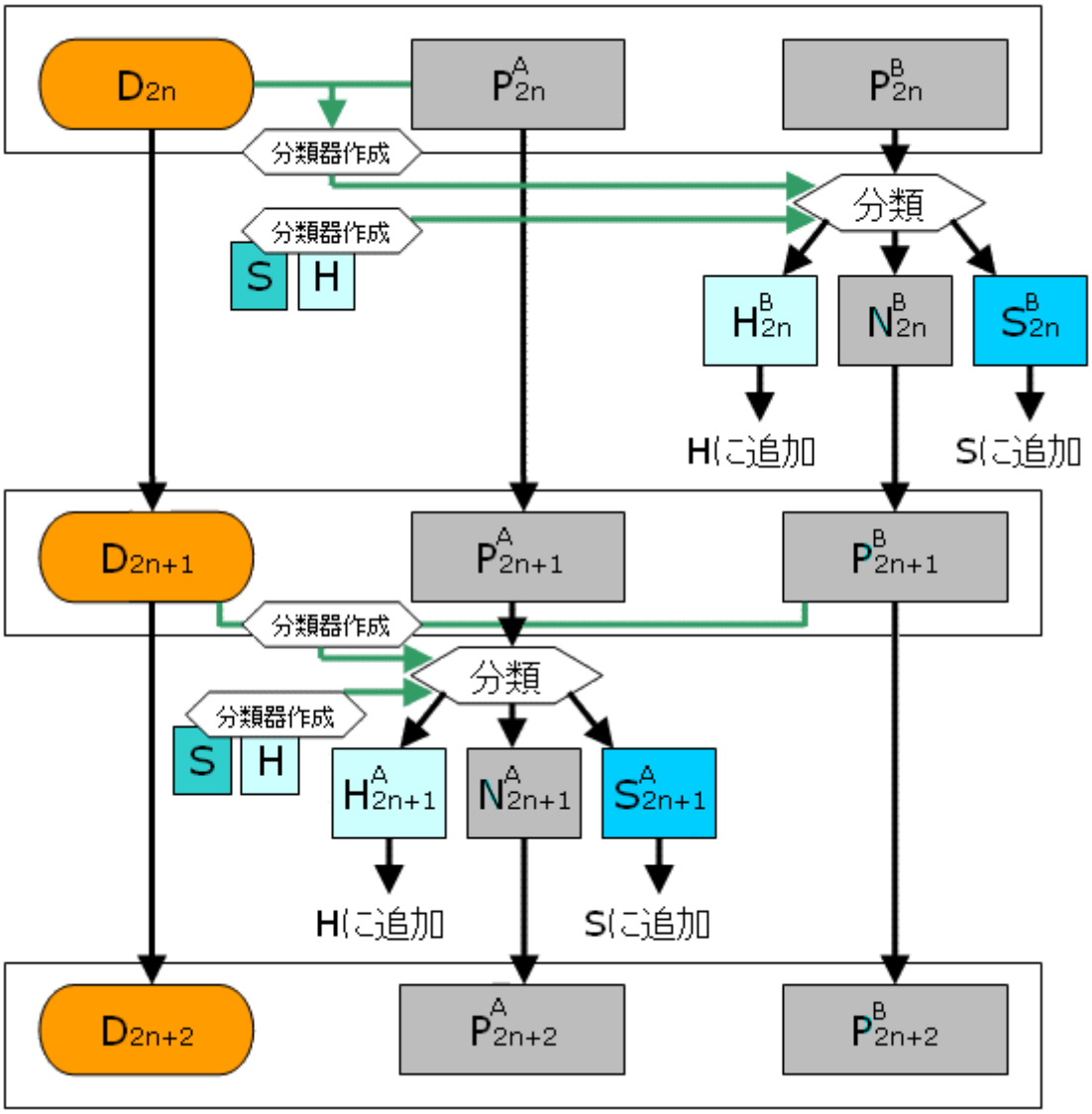

図 3.2 スパム・ハム分離型のコーパス精錬方法

3) スパム・ハム分離型の精錬手法

不適切な文書のことはスパムと呼ばれるが，それとは逆の適切な文書はハムと呼ばれている．この手法は，未知コーパスからスパムのみを分離するのではなく，ハムも分離することによって，さらにコーパスの分類精度を上げようとするものである．この手法によるアルゴリズムは以下のようになる．

1. コーパスから，教師つき負例データセット D_0，２組の未知データセットそれぞれ P_0^A と P_0^B を抽出する．
2. n は $n \geq 0$ の整数とする．D_{2n} と P_{2n}^A から文書を１：１にランダムに抽出して作成された SVM 学習器と，S と H から作成された SVM 学習器を 2 つ用いて P_{2n}^B を分類する．この 2 つの SVM 学習器両方によってスパムだと判定されたもののうちある条件 Q を満たすものを S_{2n}^B とする．また，SVM 学習器によって出力されたもののうち, スパムでない符号で最も絶対値が大きいほうから S_{2n}^B の文書数$|S_{2n}^B|$と同じ件数だけをハム H_{2n}^B とし，同じく P_{2n}^B から分類する．残りを N_{2n}^B とする．
3. $D_{2n+1} = D_{2n}$ ， $P_{2n+1}^A = P_{2n}^A$， $P_{2n+1}^B = N_{2n}^B$ とする．
4. D_{2n+1} と P_{2n+1}^B から文書を１：１にランダムに抽出して作成された SVM 学習器と，S と H から作成された SVM 学習器 2 つを用いて P_{2n+1}^A を分類する．これによってスパムだと判定されたもののうちある条件 Q を満たすものを S_{2n+1}^A とする．2.と同様にハム H_{2n+1}^A も分離し，残りを N_{2n+1}^A とする．
5. $D_{2n+2} = D_{2n+1}$ ， $P_{2n+2}^A = N_{2n+1}^A$， $P_{2n+2}^B = P_{2n+1}^B$ とする．

6.2.～5.を，S として取り除かれる投稿文が収まるまで繰り返す．

スパムとして取り除かれた文の集合を $S = S_0^B + S_1^A + S_2^B + S_3^A + \cdots$，ハムとして取り除かれた文の集合を $H = H_0^B + H_1^A + H_2^B + H_3^A + \cdots$ と定義する．アルゴリズムによる要請により，$|H_{2n}^B| = |S_{2n}^B|, |H_{2n}^B| = |S_{2n}^B|$ になるので，常に $|H| = |S|$ となる．また，任意の n において $D_n = D_0$ となる．

3.4 実験

3.4.1 コーパス

本研究では，Yahoo! 知恵袋の一月分の投稿データと，削除された投稿データの提供をいただいた．人手により削除された投稿データは，Yahoo!知恵袋のスタッフがチェックして不適切であると判定されたものである．今回は削除された投稿の約 85%を占めていた質問文書を分析対象とする．一月分の投稿データにおける適切な質問文書（215,288 投稿）と削除された質問文書（33,852 投稿）をコーパスとした．質問投稿文の長さは平均して 60～70 文字であった．

3.4.2 評価尺度

本研究では，不適切な投稿(スパム)をいかにして取り除くかということが重要である．このため，生成されたコーパスを分類尺度として不適切投稿の分類精度に着目した次の評価尺度を用いる．

$$\text{Spam}-\text{Recall} = SS/(SS+SN)$$

$$\text{Spam}-\text{Precision} = SS/(SS+NS)$$

$$\text{Spam}-\text{F値} = \frac{2*\text{Spam}-\text{Recall}*\text{Spam}-\text{Precision}}{\text{Spam}-\text{Recall}+\text{Spam}-\text{Precision}}$$

$$\text{Spam}-\text{Accuracy} = (SS+NN)/(SS+NS+SN+NN)$$

SS = スパム文のうち，スパムだと分類された投稿文の数

SN = スパム文のうち，スパムだと分類されなかった投稿文の数

NS = スパム文ではないもののうち，スパムだと分類された投稿文の数

NN = スパム文ではないもののうち，スパムだと分類された投稿文の数

Spam-Recall とは，スパム文をスパムだと判定できた割合であり，Spam-Precision はスパムだと判定された文書のうち，実際にスパム文だった割合である．

表 3.1 素性の種類による分類精度

主義語	記号	文末	Spam-Precision	Spam-Recall	Spam-Fvalue	Spam-Accuracy	素性数
○			70.94%	88.52%	78.76%	76.42%	23,199
	○		62.76%	92.21%	74.69%	69.13%	1,248
○	○		72.80%	89.96%	80.48%	78.44%	24,347
		○	50.92%	95.90%	66.52%	52.33%	45
○		○	71.41%	89.55%	79.46%	77.13%	23,398
	○	○	65.18%	92.83%	76.59%	71.96%	1,293
○	○	○	73.13%	90.37%	80.84%	78.85%	24,646

3.4.3 素性の種類による分類精度の変化

まず素性の種類によって分類精度にどのような違いが出るかを見た．学習データとしてスパム文 10,000 投稿とスパムでない文 20,000 投稿，テストデータとしてはヤフー株式会社の方にご協力いただいて確かにスパム/スパムではないと判別した 988 投稿(うちスパムが 500，スパムでないものが 488)を用いる．

特徴ベクトルは，第 2 章と同様に各素性の *tfidf* 値を用いた．(2.3.2 節参照)素性の種類は，単語単体で意味を持つと考えられる主義語(名詞，動詞，形容詞)，記号，そして文末表現と考えられる単語(助動詞，終助詞)である．これらの単語の N-Gram(N=1～5)を抽出して，出

現回数が 3 回以上となるものを素性とした．形態素解析パーサとしては Chasen[26]を用いている．また，今回はストップワードとして半角のアルファベット 1 文字で構成される 52 の単語と，半角文字または全角文字による 1 文字の数字からなる単語 20 単語の計 72 単語を設定した．

素性の種類を様々に変えて分類した結果が表 3.1 である．これを見ると，やはり主義語が入っている場合は分類精度が高いことが分かる．次いで記号も素性として十分に機能していることが分かるが，文末表現に関してはわずかに精度が上昇するにとどまっている．これは，今回文末表現が助動詞，終助詞で構成されるものとして実験したが，実際には素性数が 45 であり，文末表現が助動詞，終助詞のみとは言えない．これは，「です」などの文末に用いられる表現が実際には動詞として解析されているケースも多いためである．ただし，この文末表現が入っていると多少は分類精度が上昇するため，今後の実験では主義語，記号，文末表現すべての品詞からなる N-Gram を素性とする．

3.4.4 学習データ件数による分類精度の変化

次に，学習データ件数を変化させたときに分類精度がどうなるのかを見てみることとした．非線形 SVM を用いるときのカーネル関数には 2 次の多項式カーネルを使用した．また，学習データの正例としてスパムでない投稿文，負例としてスパムである投稿文を用いている．テストデータの正例，負例は 3.4.3 節でも用いたテストデータ 988 投稿(うちスパムが 500，スパムでないものが 488)である．学習データの件数を変化させつつ，5 回学習と分類を行いその平均をとった．

この結果が図 3.3～図 3.5 である．図 3.3 では，学習データの総数を変化させたときの Spam-Recall の変化を示している．青い線が学習データ中の正例と負例の比率が 1:1 のときで，赤い線が学習データ中の正例と負例の比率が 2:1 のときの線である．また，ポイントが中抜きで白くなっているのが線形 SVM を用いて分類した時，ポイントが塗りつぶされているのが非線形 SVM を用いて分類したときの分類精度となっている．これを見ると，およそ学習データ数の合計が 10,000 投稿ほどで分類精度の上昇が飽和していること分かる．

図 3.4 は学習データ中の正例と負例の比率が 1:1 のときに，Spam-Recall，Spam-Precision，Spam-F 値(Spam-Fvalue)の変化を見たものである．これを見ても，学習データ数の合計が 10,000 投稿ほどで分類精度の上昇が飽和していること分かる．また，線形 SVM と非線形 SVM を比較してみると，3 種の分類尺度において非線形 SVM は線形 SVM よりも 1%ほど分類精度が高くなっていることが分かる．これにより，分類には非線形 SVM を用いたほうがよいことが分かる．

図 3.5 は 5 回の学習・分類にかかった合計時間を示している．実際には，分類にかかる時間は比較的短く，学習にかかる時間が圧倒的に大きい．これを見ると，線形 SVM は件数が増えても学習にかかる時間が大きく増えることがないのに対して，非線形 SVM ではほぼ 2 次に比例して大きく学習・分類にかかる時間が伸びている．その結果，線形 SVM に比べて

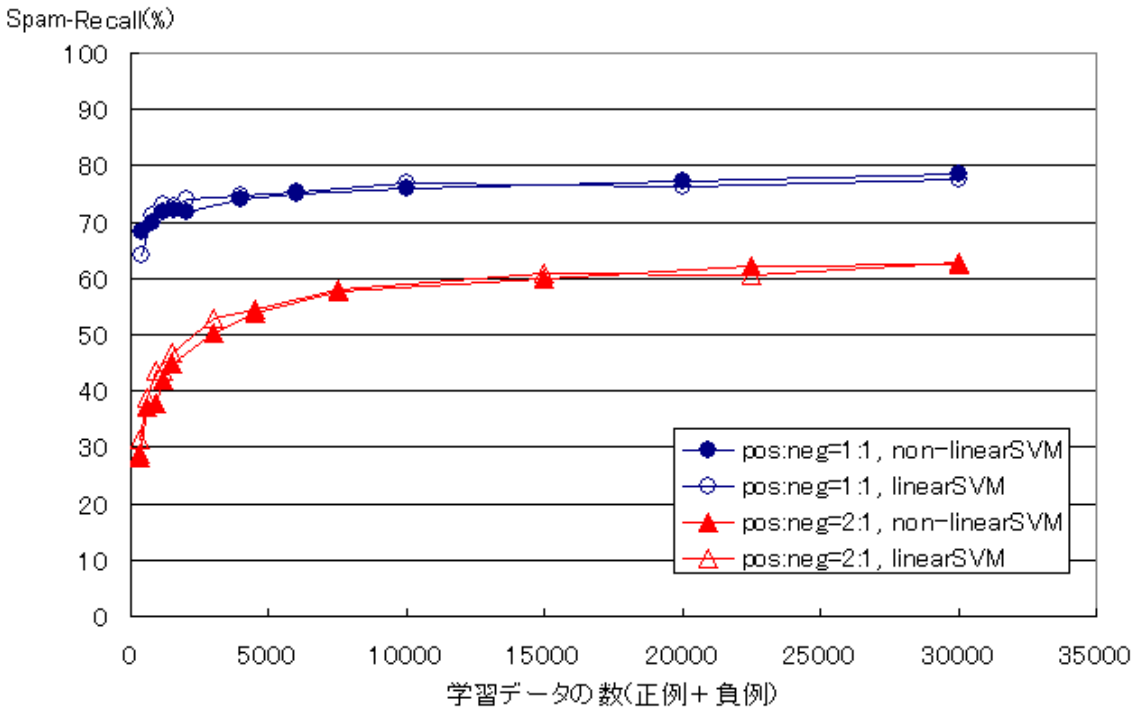

図 3.3 学習データ件数を変えたときの Spam-Recall の変化

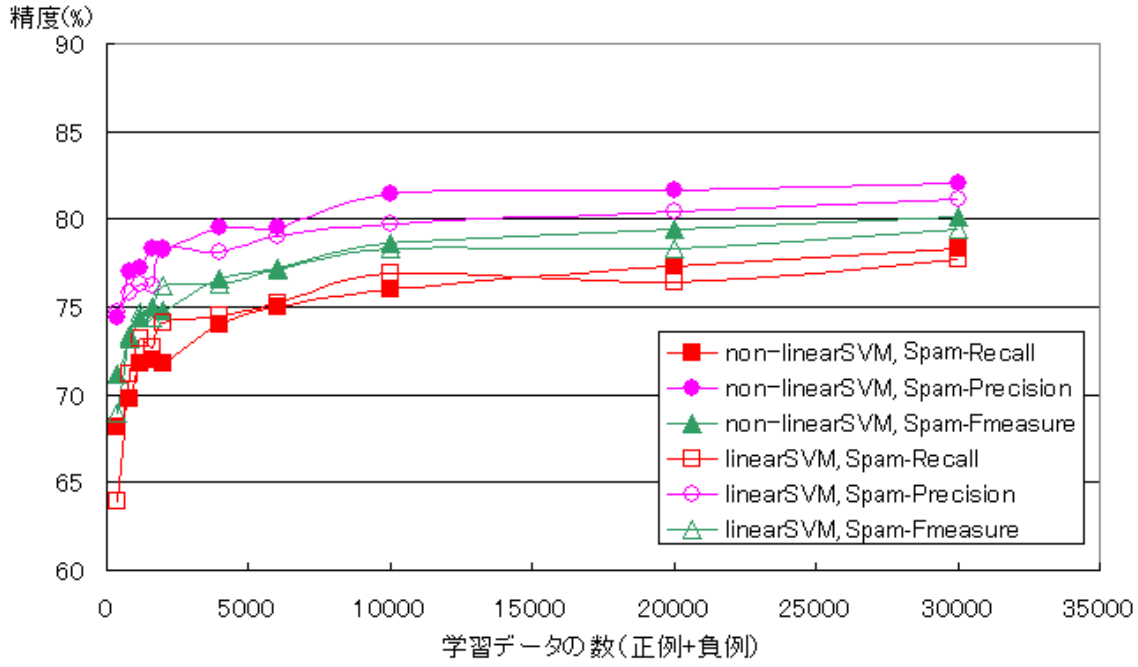

図 3.4 学習データ件数を変えたときの分類精度の変化

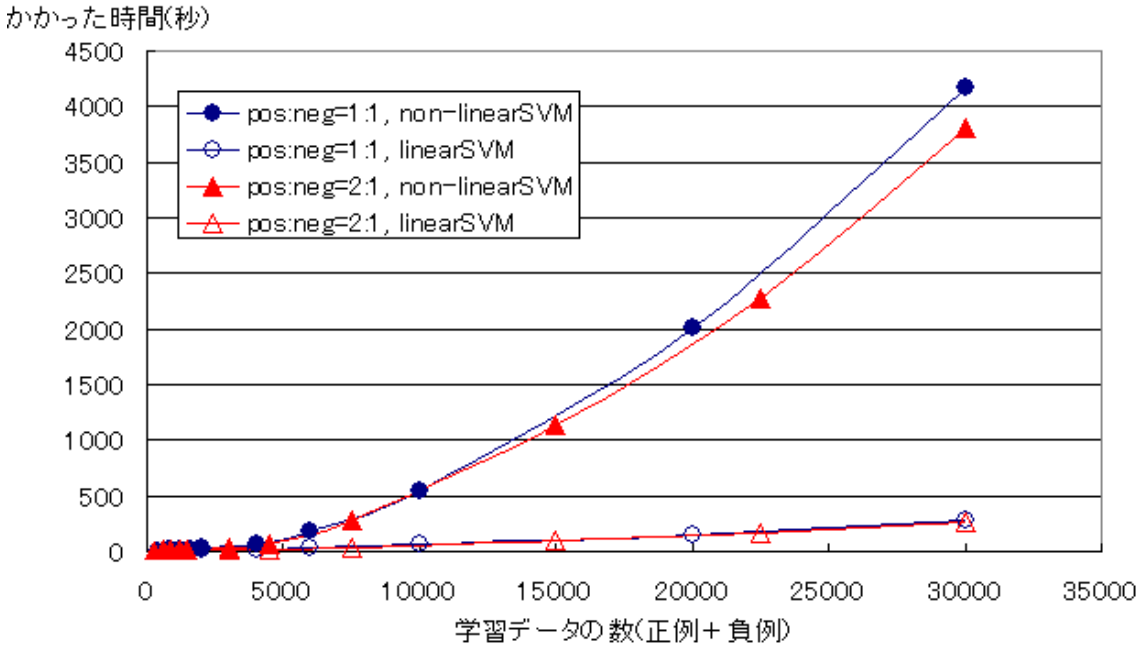

図 3.5 学習データ件数を変えたときの学習時間の変化

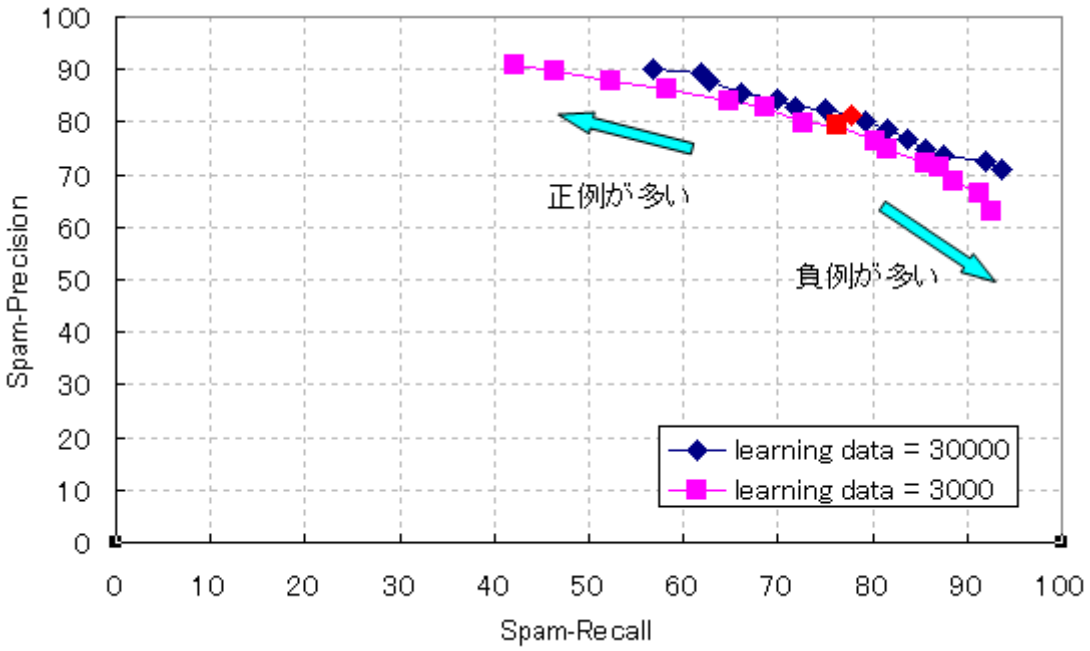

図 3.6 学習データにおける正例負例の割合を変えたときに
得られる Precision-Recall カーブ

非線形 SVM は学習・分類にかかる時間がかなり長いことが分かる．今回解析に使用した PC の CPU は Core Solo1GHz であるが，学習データ数を 30,000 としたとき，5 回の学習・分類におよそ 1 時間かかった．

また，図 3.6 は学習データ中の正例・負例を変化させたときに得られる Precision-Recall カーブを示している．正例：負例が 8:22～22:8 までをグラフにプロットしており，赤く表示されている点が正例負例比 1:1 のところを表している．また，凡例 learning data は，学習データ数の合計の数を表している．

これを見ると，正例負例比が 1:1 のところでは Spam-Precision と Spam-Recall がおよそ同じ値になり，学習データ中の負例の割合を増やした場合に Spam-Recall が上がり Spam-Precision が下がることが分かる．また，学習データ中の正例の割合を増やした場合には Spam-Precision が上がり Spam-Recall が下がっており，Spam-Precision と Spam-Recall はトレードオフの関係になっていることがわかる．

3.4.5 素性選択による分類精度の変化

3.4.4 節の結果より，分類には非線形 SVM を用いるのが望ましいが，学習・分類にかかる時間がかかりすぎるのが難点である．そこで，24,646 の素性を減らして学習にかかる時間を減らすことができないか見ることとした．素性選択(Feature Selection)の手法としては，第 2 章で用いた Brank らの手法[10]による，線形 SVM 分類器を作成したときの素性の重みを利用することとした．このときある閾値を設定して，重みの絶対値がそれ以上のもののみを素性として用いることとする．

この実験結果が図 3.7～図 3.11 である．閾値がなしとなっているのは，素性選択を行わなかったときという意味である．図 3.7 では，素性選択を行った際の素性の種類数の変化を示している．これによると，閾値がないときに 24,646 であった素性数は，閾値を 1.0 に設定したときには 912 となることが示されている．なお，閾値が 2 のときの素性数は 26 である．

図 3.8 では素性選択を行い，素性数を減らしたときの Spam-Recall の変化を見ている．この図によると，閾値が 0～1 までの間では Spam-Recall はそれほど変わらないものの，閾値が 1 を超えた場合に精度が著しく下がる(特に正例負例比が 1:1 の場合)ことがわかる．図 3.9 と合わせて見ると，Spam-Precision は Spam-Recall と違ってそれほど下がらないものの，Spam-Recall が大幅に下がることによって Spam-F 値も大きく下がっていることが分かる．

図 3.10 では閾値の変化による Spam-Recall と，特徴ベクトルを作成した際に零ベクトルになる割合，いわゆるスパースネスについて示している．これを見ると，閾値が 0.9 では素性の数が 1/20 になっているのにも関わらず，スパースネスがほとんど上昇していないことが分かる．しかし，閾値が 1 を越えたところからスパースネスが大きく上がってしまっていることが分かる．

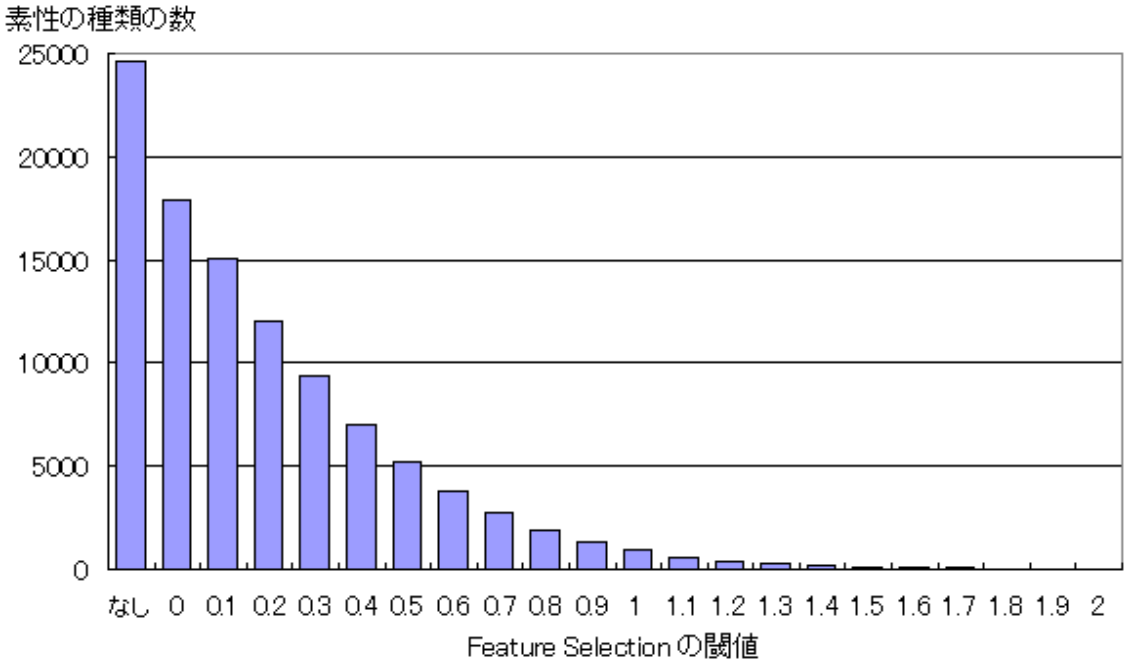

図 3.7 素性選択の閾値を変えたときの素性の種類数の変化

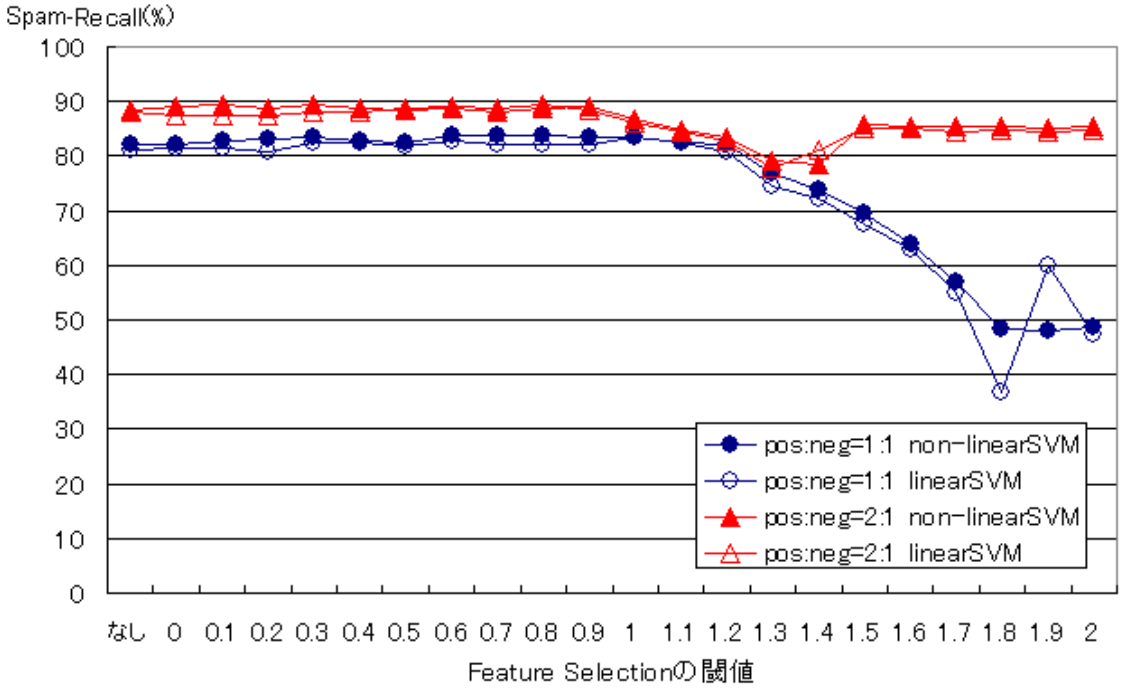

図 3.8 素性選択の閾値を変えたときの Spam-Recall の変化

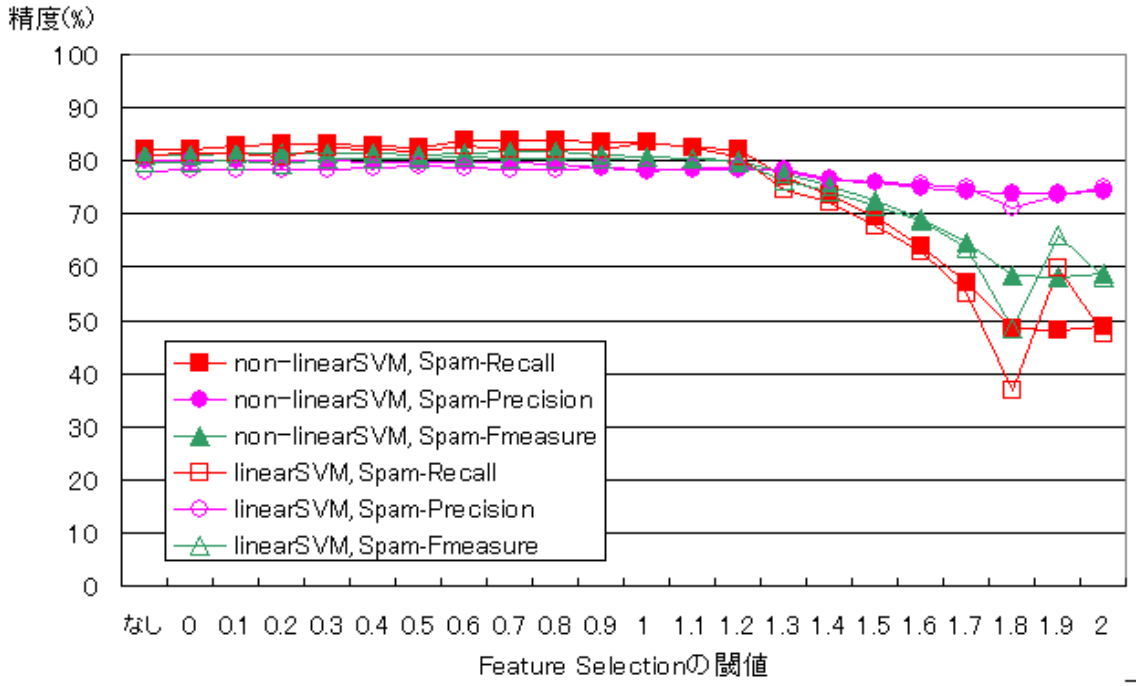

図 3.9 素性選択の閾値を変えたときの分類精度の変化

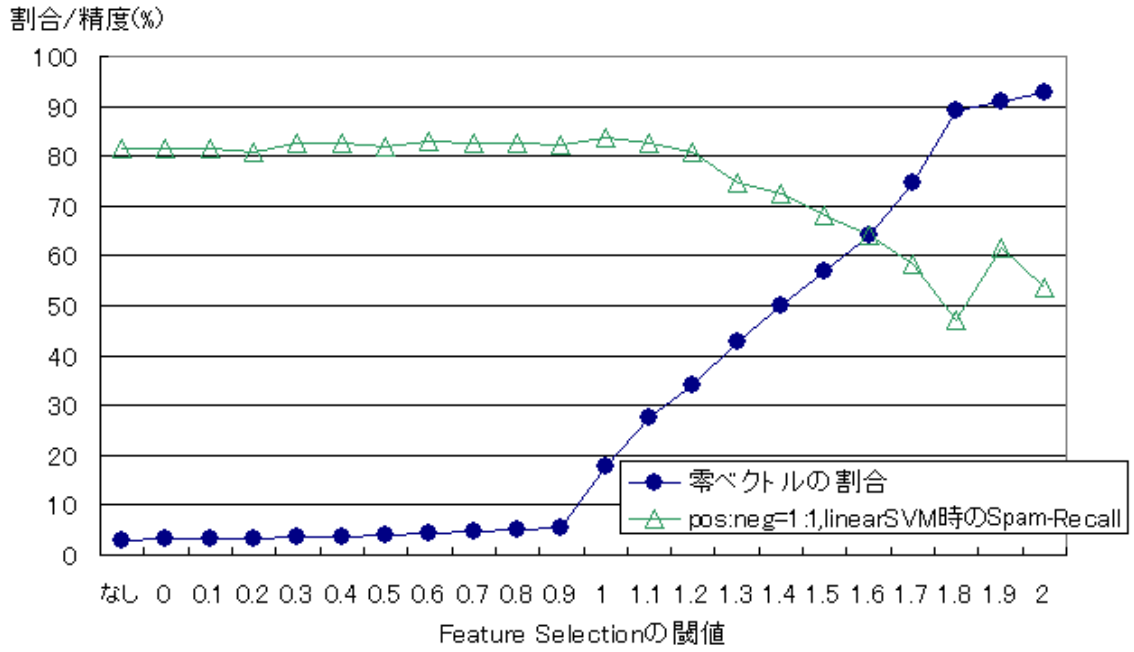

図 3.10 素性選択の閾値を変えたとき特徴ベクトルが零ベクトルになる割合

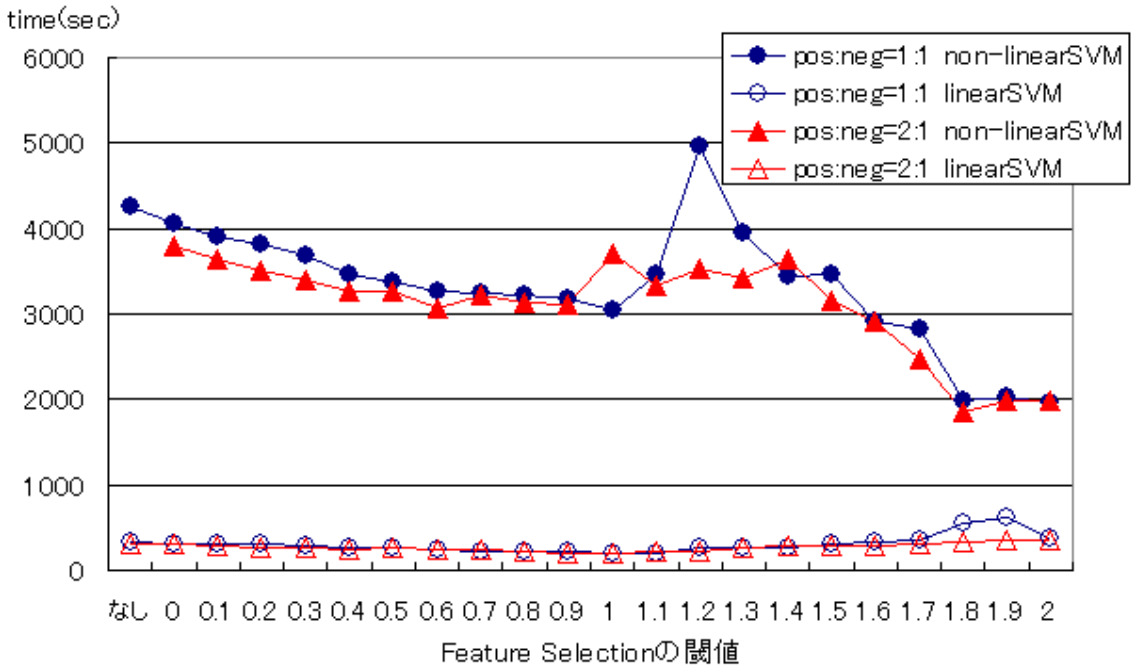

図 3.11 素性選択の閾値を変えたとき学習・分類にかかる時間

図 3.11 は SVM 分類器の学習・分類にかかる時間を示している．これも図 3.5 と同様に 5 回学習・分類を行った際の 5 回の合計時間を表している．これを見ると，素性選択は SVM 学習器の学習・分類にかかる時間を減らす効果を持っていないようである．逆に，閾値が 1.2 前後の場合は素性選択を行わない場合よりも学習・分類にかかる時間が増加している．よって，素性選択が学習・分類にかかる時間の低減には効かないことが分かった．以降では素性選択は行わないとして議論を進める．

3.4.6 条件Qによるコーパス精錬の様子

以上の準備実験を行い，コーパスの精錬に関する実験を行う．まずは，3.3.2 節のスパム分離型によって実験を行う．アルゴリズム中の条件Qをどうするかを調べる．

A) SVM によって出力が 0 より小さくなったものをすべて分離する．
B) SVM によって出力が 0 より小さくなったもののうち，値が小さい上位 100 投稿について分離する．出力が 0 より小さくなったものが 100 件に満たない場合はすべてを分離する．
C) SVM によって出力が-1 より小さくなったものをすべて分離する．

以上により実験を行う．D_0 はスパムである投稿文 5,000 投稿，P_0^A と P_0^B はそれぞれスパムである文書 8,000 投稿に加えてスパムではない文書 2,000 投稿で構成されている．

A)による結果は図 3.12 と図 3.13，B)による結果は図 3.14 と図 3.15，C)による結果は図 3.16 と図 3.17 に載せた．ここで，図 3.12，図 3.14，図 3.16 は未知コーパスの精錬状況のグラフとなっている．ここで用いられている評価尺度は，コーパスの精錬状況を表すもので, Precision とは取り除かれた文書Sのうち実際にスパムである累計の割合である. Recall はP_0^AとP_0^Bに混ざっているスパム文のうち累計で取り除くことができた割合をさしている. Fvalue はそれらを Precision と Recall とした時に求められる F 値，Accuracy はP_n^AとP_n^Bに残っているスパムではない文書とSに含まれているスパムな文書の割合を表している. 1-Precision は，Precision が累計の割合であるので，精錬 1 回で取り除かれた文書のうち実際にスパムである割合を示した. 図 3.13, 図 3.15, 図 3.17 の Spam-Recall, Spam-Precision, Spam-Fvalue は 3.4.2 節と同じである．コーパスの精錬を 5 回繰り返し，その平均をグラフに示している．

A)による結果を見ると，図 3.12 で Recall が一気に上がっているため，P_0^AとP_0^Bに含まれたスパムである文書は取り除くことができていると考えられる．しかし，Precision が低いため，スパムである文書と一緒にスパムでない文書も取り除かれてしまっていると考えられる．図 3.13 で示されている通り分類精度，特に Spam-Recall も上がっておらず，コーパスの精錬手法としてはまったく適していないことが分かる．

B)による結果を見る. まずは図 3.14 で一度にP_0^AとP_0^Bから取り除かれる文書数を制限しているため，A)による結果とは異なり，初めの Recall は低く，Precision は高い．操作を繰

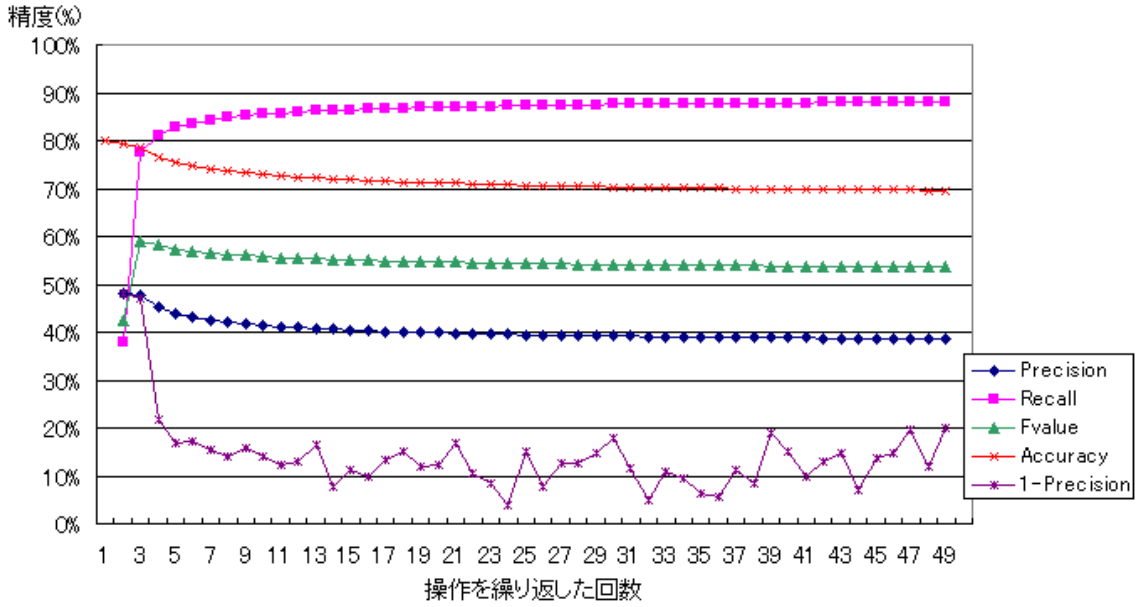

図 3.12 条件Qを A)にした時のコーパス精錬の様子

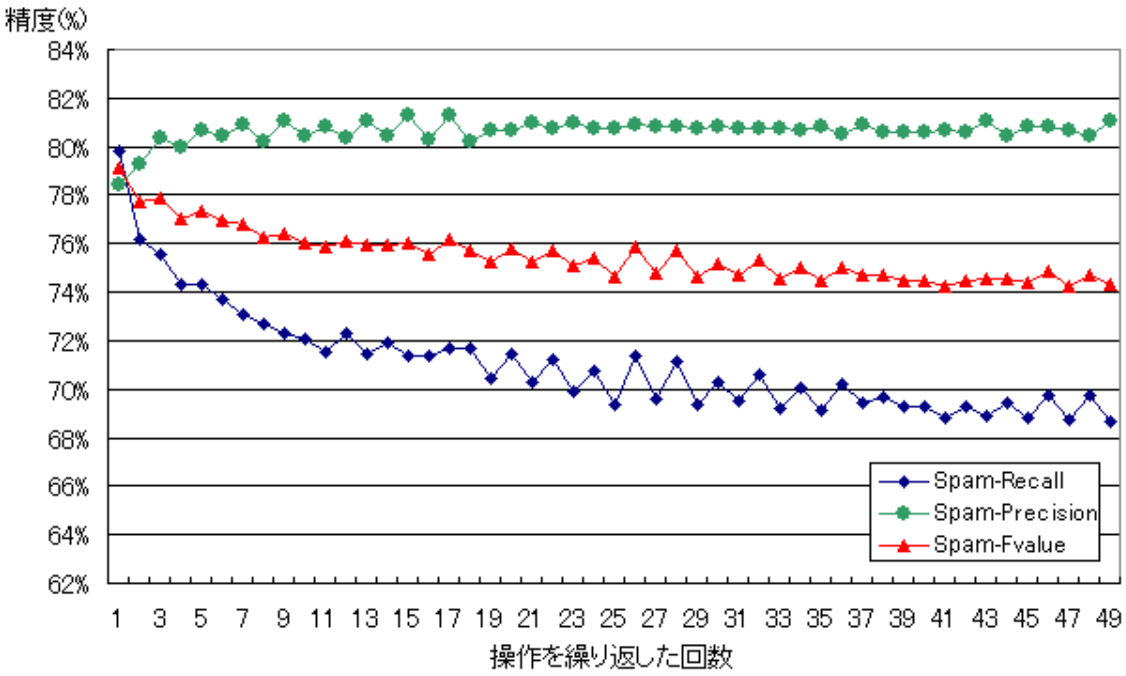

図 3.13 条件Qを A)にした時のテストデータの分類精度

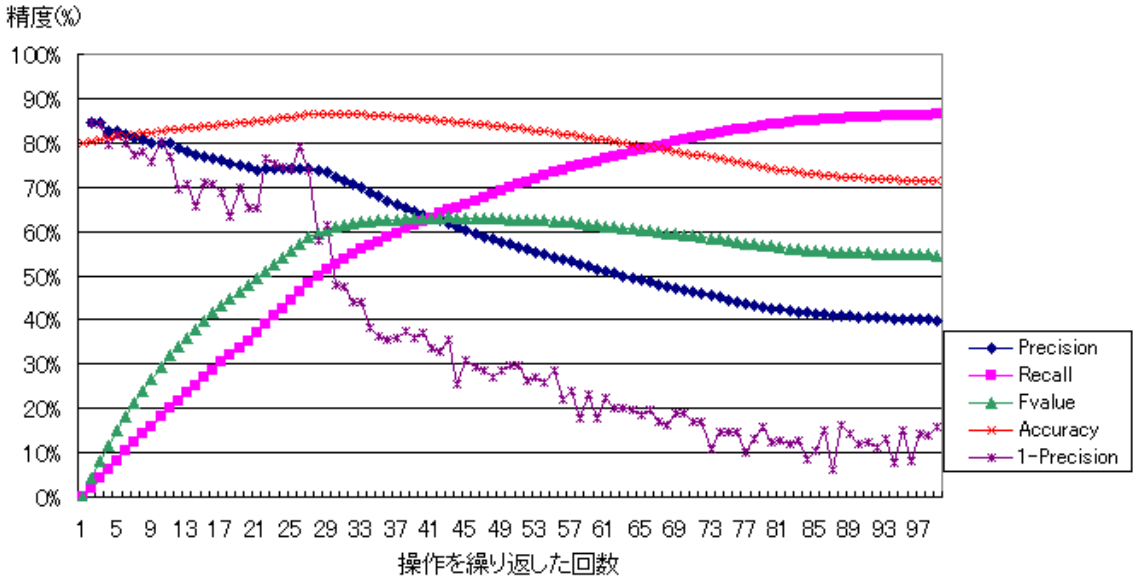

図 3.14 条件 Q を B)にした時のコーパス精錬の様子

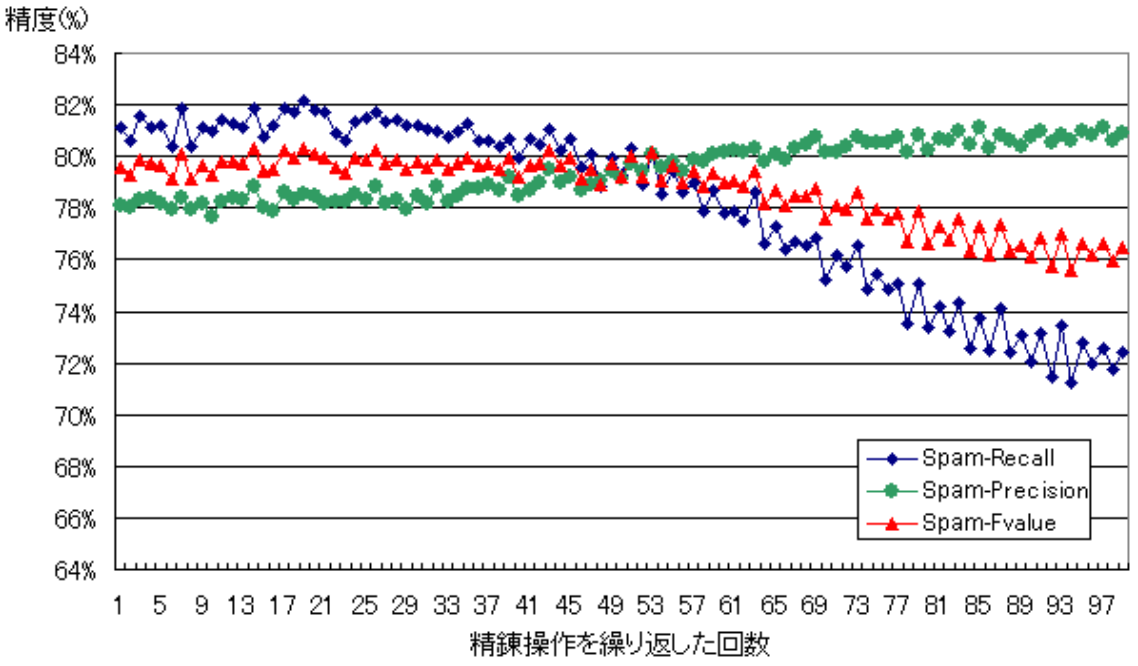

図 3.15 条件 Q を B)にした時のテストデータの分類精度

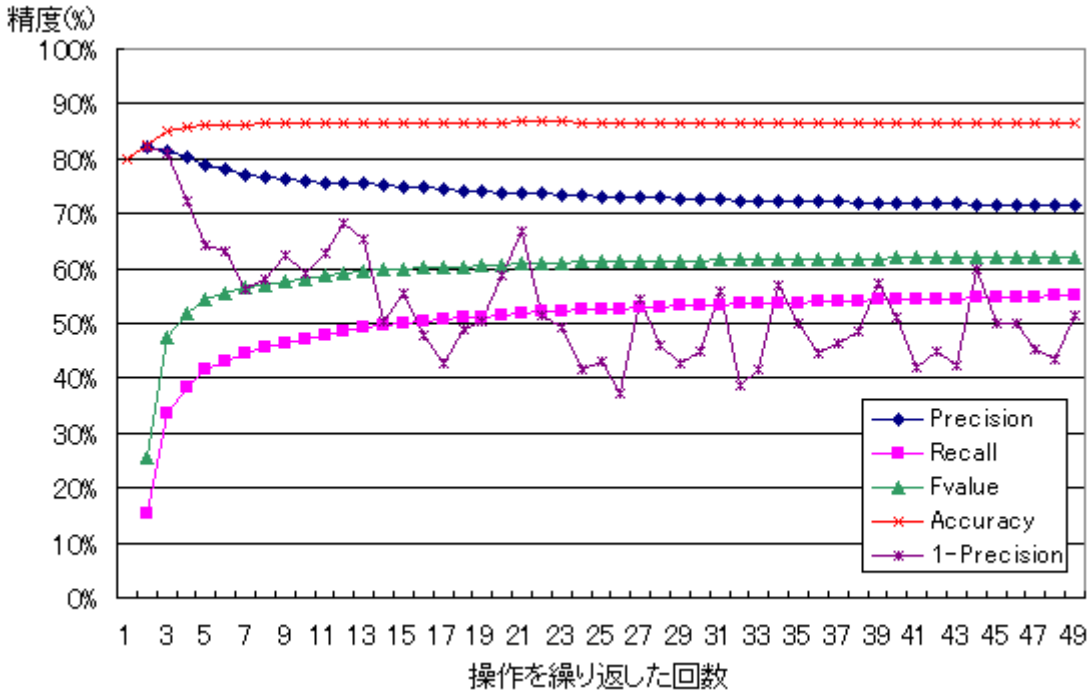

図 3.16 条件 Q を C)にした時のコーパス精錬の様子

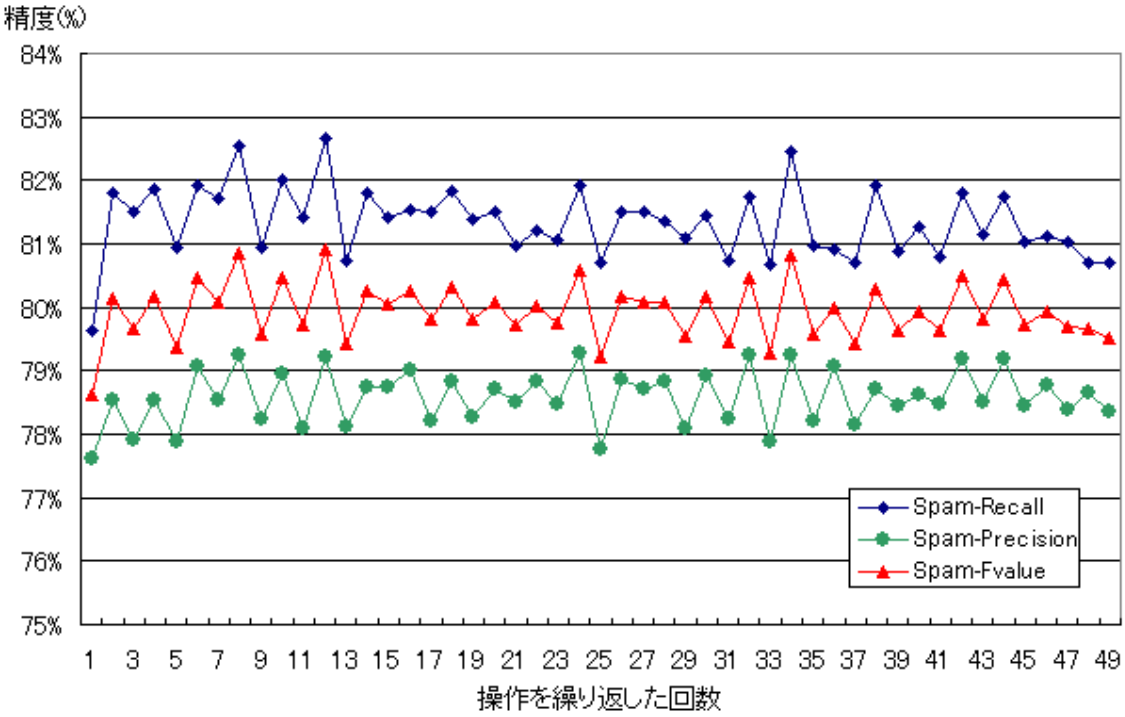

図 3.17 条件 Q を C)にした時のテストデータの分類精度

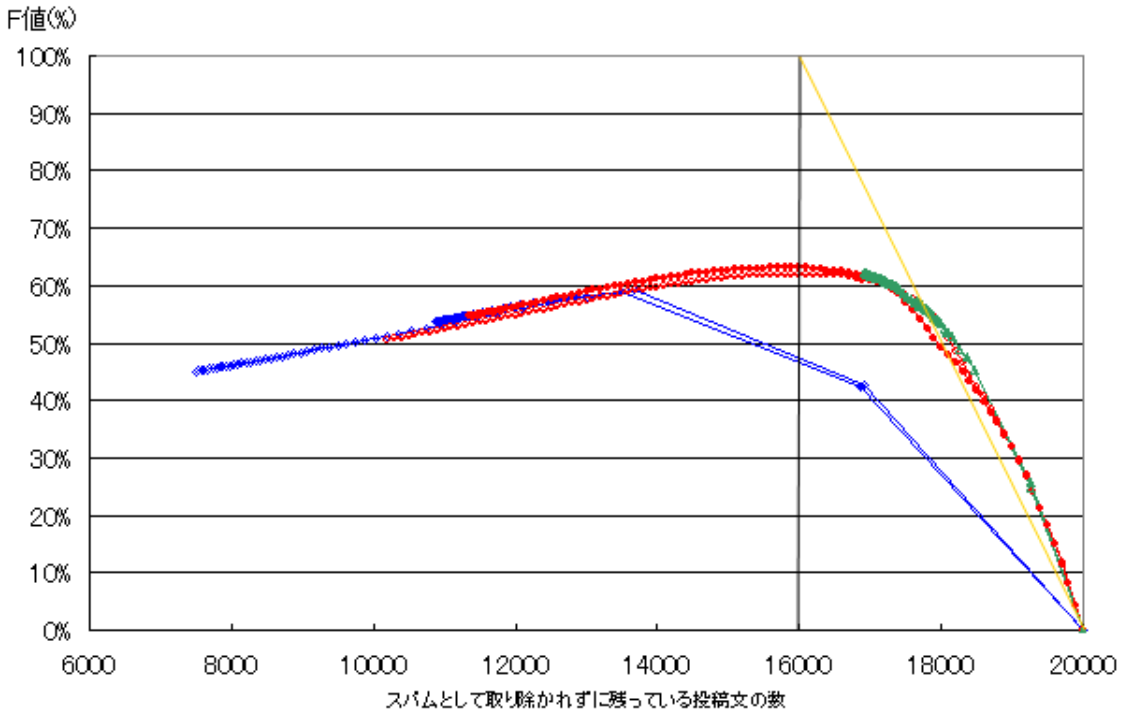

図 3.18　3 種の条件 Q の比較

り返していって，F 値の極大値が見られるので，コーパスの精錬が最もよくなる点が存在すると考えられる．しかし，そこでコーパスの精錬が収束せずに，F 値が下がっていってしまっている．そのため，図 3.15 のように途中まで上がっていったテストデータの分類精度が下がっていってしまっているという問題がある．

C)による結果の図 3.16 を見ると，初めは B)よりも勢いよく Recall が上がっているが，A)のように上がりすぎるということもなく，Precision もそれほど下がっていない．しかし，最初のみで収束してしまい，コーパス精錬の極大値に得たかどうか定かではない．これによる図 3.17 に載せてある．

この 3 種の条件Qを比較するために，図 3.18 のグラフを作った．このグラフは，X 軸にP_n^AとP_n^Bに残っている投稿文の件数$|P_n^A + P_n^B|$とY軸にその時のF値をとったグラフである．青線が A)によるもの，赤線が B)によるもの，緑線が C)によるものである．これを見ると，A)は初めから F 値の極大値を与えるところを通り過ぎてしまっていることがわかる．逆に C)は F 値の極大値を与えるところに達していない．

これにより，今後は条件Qとして B)を用いることとした．

3.4.7 アルゴリズムの違いによるコーパス精錬の様子

本研究では，3.3.2 節で様々なアルゴリズムを提案した．これによりコーパス精錬の様子，及びテストデータの分類精度のどのような違いが出るかを見ることとする．スパム分離型のアルゴリズムに対するグラフは図 3.14，図 3.15 のとおりである．これは 3.4.6 節で述べた．

負例追加型のアルゴリズムによるコーパス精錬の結果は図 3.19 および図 3.20 である．図 3.19 を見ると，スパム分離型のアルゴリズムと大差がなく，いったん F 値の極大値を与えたのち，F 値が下がっていってしまっている．しかし，図 3.20 を見ると，Spam-Recall の値はスパム分離型のアルゴリズムに比べてかなり高い値を示すことがわかる．しかし，コーパス精錬が進むにつれてこれらのテストデータの分類精度は下がっており，スパム分離型のアルゴリズムと同じ問題を抱えているということが分かる．

スパム・ハム分離型のアルゴリズムによるコーパス精錬の結果は図 3.21 および図 3.22 に示されている．図 3.21 を見ると，いったん F 値が極大値をとったあと多少下がってはいるものの，それほどの低下がなくコーパス精錬が収束していることが分かる．また，図 3.22 によると，テストデータの分類精度も下がることなく収束している．よって，これらのアルゴリズムの中で一番優れているのはスパム・ハム分離型のアルゴリズムであることが分かる．

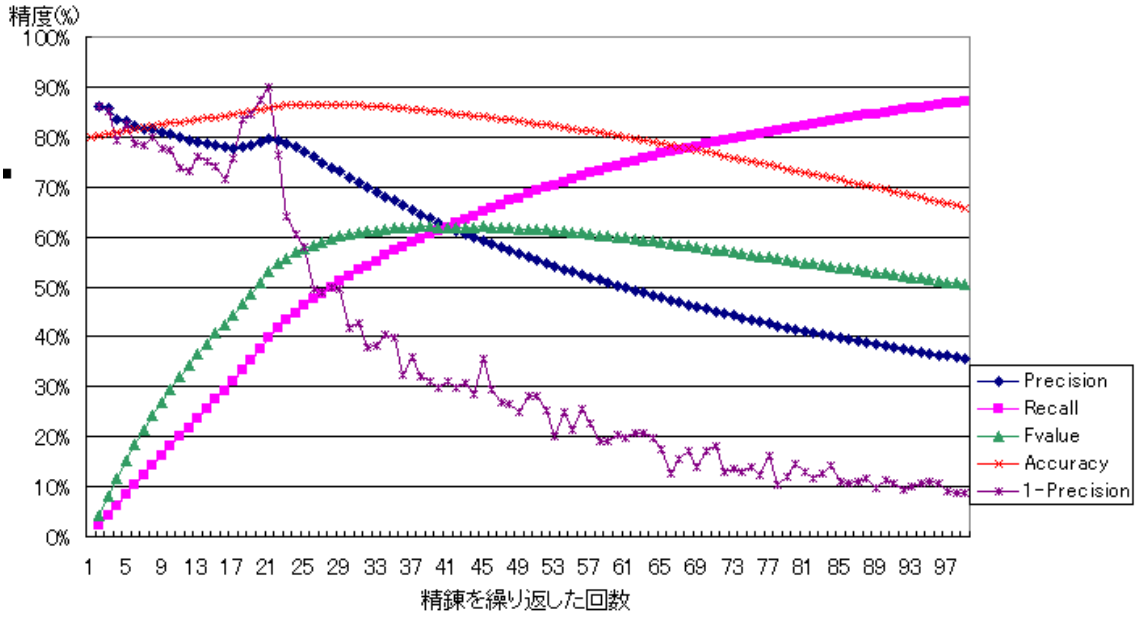

図 3.19 負例追加型アルゴリズムによるコーパス精錬の様子

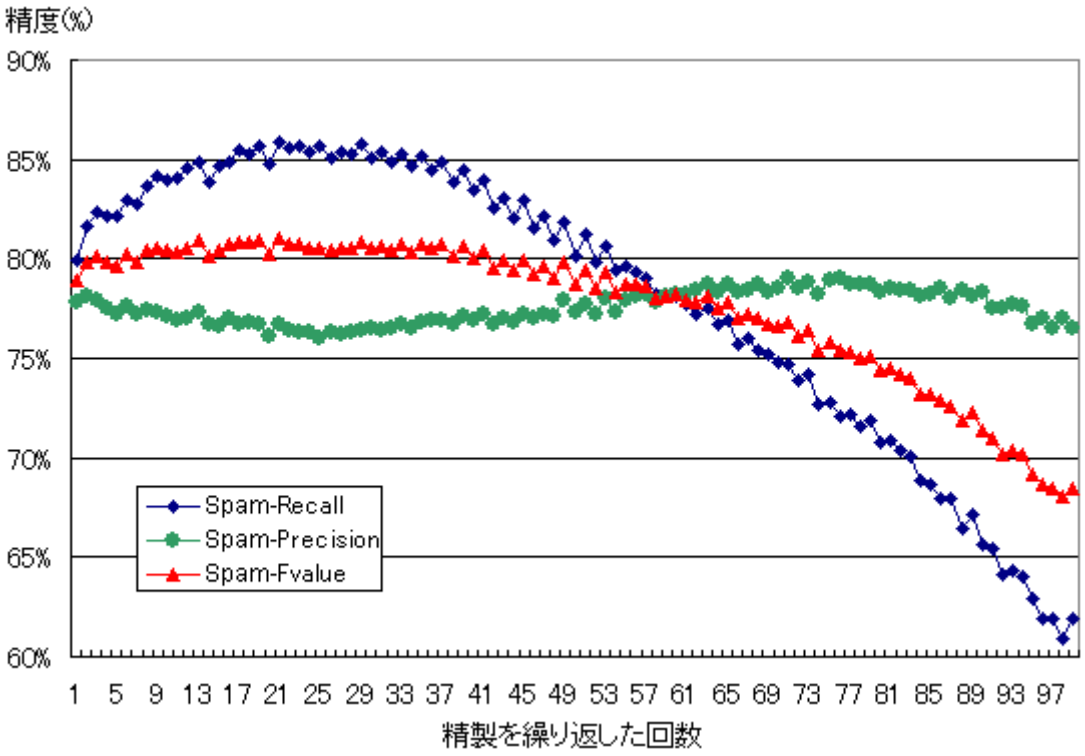

図 3.20 負例追加型アルゴリズムによるテストデータの分類精度

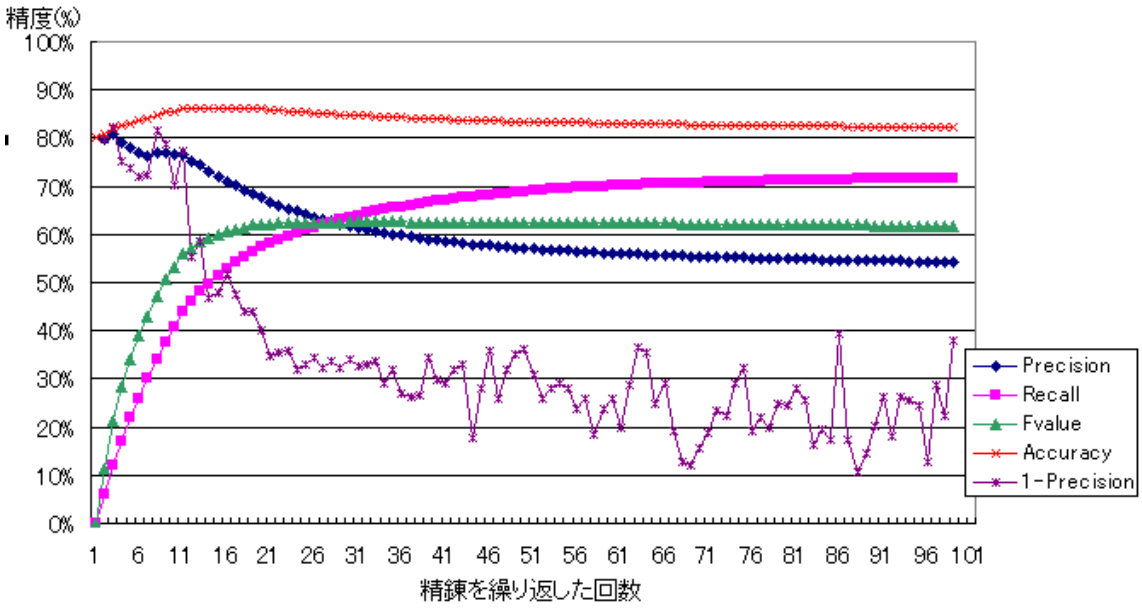

図 3.21 スパム・ハム分離型アルゴリズムによるコーパス精錬の様子

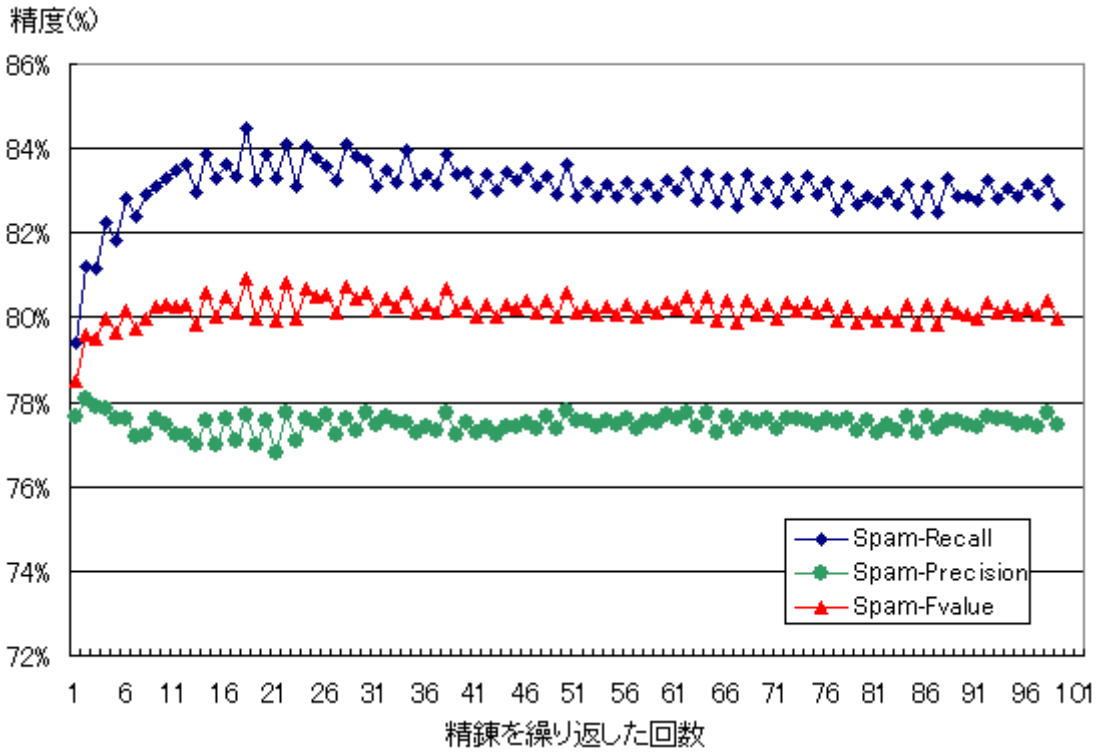

図 3.22 スパム・ハム分離型アルゴリズムによるテストデータの分類精度

3.5 議論

3.5.1 今回の実験の考察

今回の実験は，コーパスの精錬という手法を通してテストデータの分類精度が向上するかどうかを推定した．いくつかの手法では分類精度の低下しか見られなかったが，適切な手法・アルゴリズムを用いることによってテストデータの分類精度を大幅に向上させることができた.

3.5.2 関連研究と本研究の位置づけ

Web 上の不適切な投稿を自動的に分類する研究は，昔からさまざまな分野で行われてきた．有名なものとしては e-mail のスパムフィルタが挙げられる[22]．また，有害な Web サイトをブロックするための研究も行われてきた[23]. また, ブログのエントリを対象として, スパムかどうかを判定する研究にも取り組まれている [24]．しかし，本研究で対象としている不特定多数のユーザが文書を書き込むサイトに対してはあまり研究が行われていない.

一部の確かな正例・負例と，大部分の教師なしコーパスからなるコーパスから，より精度の高いコーパスを精錬して分類精度を高めようという Semi-supervised(半教師つき学習)と呼ばれる手法を用いた研究についても今までいくつか行われてきた. [27]ではテキスト分類の際に EM アルゴリズムと呼ばれる手法を用いて少数の確かな学習データと大部分の教師なし学習データをコーパスとし，少数の確かな学習データのみから分類器を生成した時よりも高い分類精度を得ることに成功している. [28]では EM アルゴリズムを用いた Naïve Bayes 分類器および SVM 分類器を作成し，ブログ上の評判情報が肯定的な表現であるか，あるいは否定的な表現であるかを分類している．これを用いた場合，EM アルゴリズムを用いた時の分類精度は EM アルゴリズムを用いなかった時よりも 0.5%～1.5%ほど向上している.

これらの研究は，コーパスに一部ではあるが正例・負例を用意した時の学習であるが，本研究の提案手法は，正例あるいは負例の片方のみ確かなコーパスが存在し，のこりは不確かなコーパスであることがこれ他の関連研究と異なる点である．さらに提案手法で 4%ほどの分類精度の向上が得られたことは，本研究が効果的な手法であることを示しているといえるだろう.

3.5.3 スパム分類以外への提案手法の適用の可能性

提案手法のコーパスの精錬は，教師つき負例と未知データからなる半教師つき学習コーパスをもとに高い分類精度を導く精錬されたコーパスを作成するという研究である．これまで，少数の教師つき正例，教師つき負例と多数の教師なし学習データを基に強化学習を行う研究はあったが，本研究のように正例負例の一方だけが教師つきであり，それ以外は

不確かである学習データを対象とした研究は前例がないようである．本研究の実験ではテストデータの分類精度もよく上昇しており，このような手法による研究も今後行われる価値はあるだろう．

このような手法の適用範囲については，今後議論の余地がある．本研究では Web における自然言語投稿文からスパム文書を分類するため，このような手法をとった．削除されたデータは確からしいが，Web における自然言語投稿文は内容・用語・書式などが様々で，削除されたデータではないとして与えられているデータであっても，スパムである文書に内容が近い文書も含まれていると考えられる．このように敢えて 2 分すればスパムではないと判断されるが，実際にはボーダーライン上にいるような文書が含まれているため，テストデータによる分類精度の低下が起こると考えられる．このため，Web における自然言語投稿文について，このような手法を用いて分類を行うのは精度を上げるのに非常に有効であると考えられる．

単に「削除されたデータ」である教師つき負例と，それ以外であるデータで学習器を作るより高いテストデータの分類精度を得られたということは，この仮説が正しいものであることを示していると言えるだろう．

第 4 章 結論

本論文によって得られた知見について以下にまとめる.

第 2 章では Web における自然言語投稿文の 1 種であるブログ記事について，教師つきのデータからブログ投稿者の性別推定を行い，84%のブログ投稿者を 90%という高い精度で分類することができた. また，提案手法により抽出される単語の重みという情報を利用し，男性もしくは女性に特徴的に用いられる単語を抽出することができた．さらに，ブログ記事単体の分類は今回の研究には，他の Web における自然言語投稿文である掲示板への投稿などへ適用することで，ブログ以外にもこの手法を用いて自然言語投稿文の書き手の性別推定ができるようになる可能性について触れた.

第 3 章では Web サイトのひとつである知識検索サイト，いわゆる QA サイトにおける自然言語投稿文の中に含まれる学習に基づくスパムの分類について述べ，コーパスの精錬という手法を通じ，高い分類精度でスパムの分類ができることを示した．この教師つき負例と教師なし正例によるコーパスの精錬という手法は前例がなく，さらに本論文の実験によって高い精度を得られることが分かったため，本論文において述べた対象に限らず用いることができる可能性があることことを示した.

本論文の 2 つの手法による Web における自然言語投稿文の学習に基づく分類では，Web の中でもユーザが手軽に文書を投稿することができるブログや知識検索サイトといったものを対象としている．これらの利用者は年々拡大しており，Web の増え続ける膨大な知識のかなりの部分を占める．これらの自然言語投稿文について学習に基づく分類を行うことによって，Web における自然言語投稿文からの知識の抽出もより容易に行われるようになっていくだろうと考えている.

謝辞

本研究を行うに際しまして，皆様にたくさんのご指導・ご鞭撻をいただきましたこと感謝いたします.

指導教官であります石塚満教授には忙しい時間を縫って日ごろからご指導･ご鞭撻を受けさせていただきました．心より感謝いたします.

石塚研究室秘書の藤田メイコさんには日ごろから快適に研究を行うための環境を整えてくださいました．こちらにも心より感謝いたします.

石塚研究室助手の土肥浩氏には，本業のかたわら石塚研究室のサーバやネットワークの管理などの研究環境の整備，そしてミーティングでの助言など，様々な面においてお世話になりました．あわせて感謝いたします.

石塚研究室の OB で，現在大阪大学大学院経済学研究科講師の松村真宏氏には，本研究に対して広範囲に多大な尽力をいただき，また数多くのご指導･ご鞭撻をいただきました．ここに心よりの感謝の意を示します.

ヤフー株式会社の木戸冬子様には第 3 章の研究に関して数多くのご指導，そして数多くのご鞭撻をいただきました．ここに記して感謝いたします.

石塚研究室の OB で，現在産業技術総合研究所研究員の松尾豊氏には，研究に対しての様々なアドバイスなど数多くのご指導･ご鞭撻をいただきました．心より感謝いたします.

博士 3 年の岡崎直観氏，また森純一郎氏には研究生活の中で様々なアドバイスをいただき，また日々の雑談などもさせていただき様々な面でお世話になりました．ここに記して感謝いたします.

修士 2 年のボッレーガラ　ダヌシカ　タルパティ氏には学部 4 年から研究室が一緒だったということもあり，研究のアドバイスや日々の会話などでお世話になりました．ここに記して感謝いたします.

石塚研究室の先輩，後輩の皆様には研究における指導など非常に多くの経験をいただきました．ここの感謝の意を記します.

第 2 章「ブログへの投稿データを基とするブログ投稿者の性別自動推定」では，Doblog のデータ記事及び,「Doblog の利用に関するアンケート調査」のデータのご提供をいただき，分析に利用させていただきました．ご協力していただきました株式会社 NTT データ様，株式会社ホットリンク様にはここに記してお礼を申し上げます.

また，第 3 章「知識検索サイトへの不適切投稿の自動推定」では，ヤフー株式会社様より Yahoo! 知恵袋のデータの提供をいただき，そのデータを分析に利用させていただき，また，実験についてのご協力もいただきました．ご協力いただきましたヤフー株式会社様には記して重ね重ねお礼を申し上げます．

最後に，学部 4 年からの 3 年間，卒業研究，修士研究を石塚研究室で行うことができたことは必ず将来にわたっての財産になることでしょう．皆様本当にありがとうございました．

参考文献

[1] 総務省「ブログ・ＳＮＳ（ソーシャルネットワーキングサイト）の現状分析及び将来予測」http://www.soumu.go.jp/s-news/2005/050517_3.html (2005)

[2] 総務省 報道資料 「ブログ及び SNS の登録者数(平成 18 年 3 月末現在)」 http://www.soumu.go.jp/s-news/2006/060413_2.html (2006)

[3] Yahoo! 知恵袋 http://chiebukuro.yahoo.co.jp/

[4] 教えて！goo http://oshiete.goo.ne.jp/

[5] 近藤 泰弘，近藤 みゆき．「平安時代古典語古典文学研究のための N-gram を用いた解析手法」言語情報処理学会第 7 回年次大会『発表論文集』(2001)

[6] Moshe Koppel, Shlomo Argamon and Anat Rachel Shimoni. Automatically categorizing written texts by author gender. Library and Linguistic Computing, Col.17, No.4, 2003.

[7] Malcolm Corney, Olivier de Vel, Alison Anderson and Georoge Mohay. Gender-preferential text mining of E-Mail discource. In 18th Annual Computer Security Applications Conference, 2002.

[8] Rada Mihalcea and Hugo Liu. A corpus-based approach to finding happiness. AAAI Spring Symposium on Computational Approaches to Weblogs, March 2006.

[9] Pranam Kolari, Tim Finin and Anupam Jochi. SVMs for the Blogsphere: Blog Identification and Splog Detection. AAAI Spring Symposium on Computational Approaches to Weblogs, March 2006.

[10] Janez Brank, Marko Grobelnik, Nataša Milic-Frayling and Dunja Mladenic. Feature Selection Using Linear Support Vector Machines. Proc. of the 3rd Int. Conf. on Data Mining Methods and Databases for Engineering, Finance, andOther Fields, Bologna, Italy, September 2002.

[11] C. Cortes and V. Vapnik. Supports-vector networks. Machine Learning/ 20:273-297, November 1995.

[12] P. Eckert. Gender and sociololinguistic variation. J.Cortes., Readings in Language and Gender, pages 64-75, 1997.

[13] S. Herring. Two variants of an electronic message schema. In S. Herring ed., Computer-Mediated Communication: Linguistic, Social and Cross-Cultural Prespectives, pages 81-106, 1996.

[14] J. Holmes. Women's talk: The question of sociolinguistic universals. Australian Journal of Communications, 20(3), 1993.

[15] T. Joachims. Text categorization with support vector machines; Learning with many relevant features. In Proc. Europia Conf. Machine Learning, pages 137-142. ECML, 1998

[16] S. Nowson and J. Oberlander. The identity of bloggers: Openess and gender in personal weblogs. In AAAI Spring Symposium Series Computational Approaches to Analyzing Weblogs, pages 163-167, March 2006.

[17] M. Palander-Collin. Male and female styles in 17th century correspondence: I think. Language Variation and Change, 11:123-141, 1999.

[18] J. Schler, M. Koppel, S. Argamon and J. Prennebaker. Effects of age and gender on blogging. In AAAI Spring Symposium Series Computational Approaches to Analyzing Weblogs, pages 199-205, March 2006.

[19] J. Simkins-Bullock and B. Wildman. An investigation into relationship between gender and language. Sex Roles, 24, 1991.

[20] V. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, New York, 1995.

[21] X. Yan and L. Yan. Gender classification of weblog authors. In AAAI Spring Symposium Series Computational Approaches to Analyzing Weblogs, pages 228-230, March 2006.

[22] H. Drucker, C. Wu and V. Vapnik, Support Vector Machines for Spam Categorization. IEEE Trans. On Neutral Networks, vol. 10, number 5, pp.1048-1054, 1999.

[23] B. Grailheres, S. Brunessaux, P. Leray, Combining Classifiers for harmful document filtering, RIAO'2004, Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval, 2004.

[24] P. Kolari, A. Java and T. Finin, Characterizing the Splogosphere. Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wide Web Conference, 2006.

[25] Yahoo! 知恵袋ガイドライン http://chiebukuro.yahoo.co.jp/docs/guidelines.html

[26] 松本裕治，北内啓，山下達雄，平野善隆，松田寛，高岡一馬，浅原 正幸：日本語形態素解析システム『茶筌』 version 2.2.1 使用説明書(2000)

[27] K. Nigam, A. Mccallum, S. Thrun and T. Michell. Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning, 39(2/3). pp. 103-134. 2000.

[28] 鈴木 泰裕, 高村 大也, 奥村 学. "Semi-Supervised な学習手法による評価表現分類", 言語処理学会 第 11 回年次大会， pp. 668--671 , March 2005.

[29] 財団法人インターネット協会，“インターネット白書 2004”，インプレス，2004.
[30] Doblog http://www.doblog.com/
[31] SVMlight http://svmlight.joachims.org/

発表文献

[1] 小林 大祐，谷口 智哉，石塚 満：“Web テキスト文からのルール知識の抽出”，第 67 回情報通信学会全国大会講演論文集 5R-1(2005)

[2] 小林 大祐，松村 真宏，石塚 満：“ブログ記事の書き手の男女分類”，言語処理学会第 12 回年次大会ワークショップ「感情・評価・態度と言語」論文集 pp.73-76(2006)

[3] 小林 大祐，松村 真宏，石塚 満：“blog の文書を元にした文書の男女分類”，電子情報通信学会 第二種研究会資料 [Web インテリジェンスとインタラクション] WI2-2006-14 pp.13-18(2006)

[4] 小林 大祐，松村 真宏，石塚 満：“知識検索サイトにおける有害情報のフィルタリング知識の表出化”，第 20 回人工知能学会全国大会 1A1-03(2006)

[5] 小林 大祐，松村 真宏，石塚 満：“ブログ分類知識に基づく男性語・女性語の抽出”，第 20 回人工知能学会全国大会 3D2-04(2006)

[6] 小林 大祐，松村 真宏，木戸 冬子，石塚 満：“知識検索サイトにおける不適切な投稿の分類”第 67 回情報通信学会全国大会後援論文集 5ZB-2(2007)

[7] D. Kobayashi, N. Matsumura and Mitsuru Ishizuka: Automatic Estimation of Bloggers' Gender. In International Conference on Weblogs and Social Media, March 2007.