原著論文

日本語学術論文 PDF ファイルの自動判定

Automatic identification of academic articles in Japanese PDF files

安形 輝
Teru AGATA
池内 淳
Atsushi IKEUCHI
石田栄美
Emi ISHIDA
野末道子
Michiko NOZUE
久野高志
Takashi KUNO
上田修一
Shuichi UEDA

Résumé

As open-access policies gain acceptance, an increasing number of researchers are contributing their papers to publicly accessible web sites (i.e. self-archiving). Theoretically, these papers are accessible from standard search engines, but they tend to be obscured by other contents on the web. The purpose of this research is to develop a system that can automatically detect

安形 輝：亜細亜大学，東京都武蔵野市境 5–24–10
Teru AGATA: Asia University, 5–24–10 Sakai Musashino-shi, Tokyo
e-mail: agata@asia-u.ac.jp
池内 淳：大東文化大学
Atsushi IKEUCHI: Daito Bunka University
石田栄美：駿河台大学
Emi ISHIDA: Surugadai University
野末道子：鉄道総合技術研究所
Michiko NOZUE: Railway Technical Research Institute
久野高志：作新学院大学
Takashi KUNO: Sakushingakuin University
上田修一：慶應義塾大学
Shuichi UEDA: Keio University
受付日：2006 年 5 月 15 日 受理日：2006 年 9 月 4 日

academic articles and/or quasi-academic articles on the web. This paper describes experiments that were conducted on the performance of various classifiers and the results are compared in terms of precision, recall, and F-measure. The classifiers use attributes such as terms in PDF files and empirical rules. The results suggest the efficiency of a ranked output system which has several phases to identify academic articles.

I. はじめに

A. オープンアクセスとは

近年，学術情報流通において，オープンアクセスに対する関心が高まっている。2001 年に開催されたオープンアクセスに関する会議の成果である「Budapest Open Access Initiative (BOAI)」による表現を借りるならば，オープンアクセスとは，“完全に無償で制約のないアクセスによって，学術文献を世界規模で電子的に提供すること”[1] とされている。ただし，オープンアクセス論者の間でも，査読付き学術論文に限定するかどうか，雑誌出版後の猶予期間を認めるかどうかなどの点において定義は一様ではない。

オープンアクセスの実現方法として一般に，研究者がプレプリントあるいはポストプリントを自分のウェブサイトに公開する「セルフアーカイビング」，機関リポジトリなど一般の人がアクセスできるウェブサイトに登録する方式，読者は無料で読むことができる「オープンアクセスジャーナル」の提供といった方法が認められている。学術論文がオープンアクセスという形で提供されることは，研究者にとっては，著者として自らの研究成果を広範囲に流通させるための制度的基盤が確立されることになり，一方，読者として，他の研究者の研究成果を容易に利用できることにつなが

る。その結果，オープンアクセスで提供されている論文はそうでない論文よりも，被引用率が高いという研究成果が報告されている[2]ようにその効用が認められつつある。

なお，2006 年 2 月時点では，9 割以上の主要な学術雑誌が，ポストプリントあるいはプレプリントをセルフアーカイビングすることを許可している[3]。

B. オープンアクセス論文へのアクセス手段

オープンアクセスへの期待とその趨勢は明らかであるが，現状では，あらゆる学術論文がネットワーク上において無償で利用できるわけでない。とくに，言語間，及び，分野間の格差は顕著である。先進的な試みとして紹介されるものはいずれも海外の事例であり，我が国では，科学技術振興機構 (JST)[4] の「科学技術情報発信・流通総合システム」(J-STAGE)[5] がオープンアクセスジャーナルを 100 タイトル以上提供している（2006 年 2 月現在）が，全体としてみれば日本の学術雑誌の電子化は極めて遅れており，特に人文社会科学分野では書誌データベースすら完備されていない領域も少なくない[6]。

一方，仮に，多くの学術論文がオープンアクセスになったとしても，その探索，入手の問題は依然として残される。これに対しては，機関リポジトリの横断検索サービスを提供する OAIster があり，2006 年 5 月現在，639 機関から約 732 万件が登録されているが，実際に研究者に利用されているかどうかは不明である。オープンアクセスジャーナルの場合，それが利用者にとって既知の情報源であればアクセスは容易なものとなるが，著者のウェブサイトなどによって提供される資料については，Google などの一般的なサーチエンジンを用いて検索するほか手立てはない。しかしながら，実際に特定の著者，論題の学術論文をサーチエンジンで探索しようとする場合，膨大な検索ノイズに遭遇する可能性が高いことから，何らかの統一されたインターフェースの存在が期待される。

学術論文を対象としたサーチエンジンとしては，英語圏には，CiteSeer.IST[7] や Google Scholar[8] 等，代表的なものがいくつか存在する。CiteSeer.IST は，コンピュータ科学分野を中心とした限定的な収集で，規模はそれほど大きくない。CiteSeer.IST は学術文献の自動判定を行っているが，その判断は引用文献の有無による自動的な判定である[9]。Google Scholar は分野を限定はしていないが，検索結果には，学協会，学術出版社，大学図書館データベースベンダなどこれまで学術情報流通を担ってきたサイトしか含まれず，収集対象が限定されていると推測される[10]。言葉を換えれば，Google Scholar では，著者のウェブサイトに公開されたオープンアクセス論文は検索されない。

まとめると，オープンアクセス論文を対象とした大半の検索サービスは，(1) 先進的なサービスは英語圏が中心であり，そのようなサービスですら，(2) 研究者の個人のウェブサイトまで収集対象を広げてはいない。十分なアクセス手段が提供されていない現状において，ウェブ全体からオープンアクセスな学術論文を十分な精度で識別できるならば，実用的な学術情報検索システムの構築が可能となるはずである。

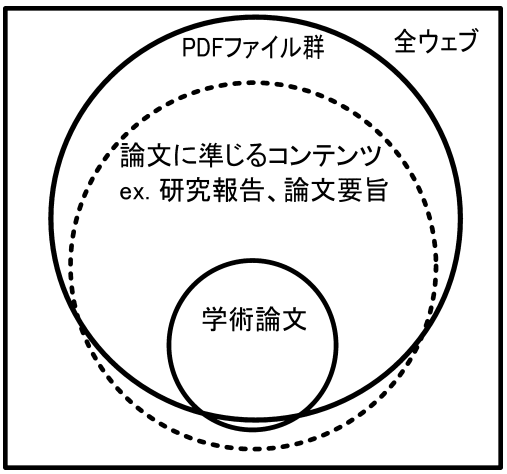
C. 日本語 PDF ファイルを対象としたウェブ上の学術情報検索エンジン

以上のことを踏まえると，ウェブ上で公開された情報から学術論文を選別し，提供する学術情報専門の検索エンジンの構築の必要性が認められよう。特に，ウェブ上の学術情報探索手段が整備されていない日本語論文を対象とした検索エンジンには需要があると考えられる。

こうした検索エンジンの実現にあたって，具体的には，まず PDF ファイルを対象とするのが妥当と考えられる。現在，学術論文の全文をファイルで提供する場合，ファイル形式には PDF, HTML, XML, TeX, MS Word などがある。なかでも，PDF 形式は他の形式と比べ文書のレイアウトやデザインを維持したまま閲覧でき，閲覧条件を設定することも可能であるため，標準的な配布形式となっている。実際に，オープンアクセス

論文に占める PDF ファイルの割合は三根慎二の調査によれば，2006 年 5 月時点で 80.9% である[11]。この状況を図にすると第 1 図のように，ウェブ上で公開されている学術論文，あるいは，それに準じるコンテンツのほとんどは PDF ファイルで提供されていると考えられる。

さらに，PDF ファイル形式ではなく，HTML ファイル形式で論文が提供されている場合に，論文の自動判定を行う場合，1 論文が必ずしも 1 ファイルで提供されるとは限らないため，1 論文となりうる範囲の判定を行う必要が出てくる。PDF ファイルに関しても，後述のように複数の論文群を 1 ファイル，あるいは 1 論文を複数ファイルに分割する場合もあるが，HTML 形式などに比べると非常に少ない。したがって，PDF ファイルが当面の対象となる。



第 1 図　ウェブ中の学術論文

さらに，ここでは，もう一つ新しい課題を設けた。それは，学術論文以外の形をとる研究報告の選別収集である。

Google Scholar など学術情報に特化した検索エンジンは，フォーマルな流通経路に乗る学術情報を扱っている。しかし，研究者同士で交換されるプレプリント，内部で作成される研究報告の資料など，インフォーマルに作成され，流通する資料の重要性は，学術情報流通の領域でたびたび指摘されてきた[12]。

ここでは学術論文の自動判定について検討していく際に，インフォーマルと位置づけられる報告をも含んだ自動判定を目指していく。ただし，インフォーマルな資料の形態はさまざまであるため，第一段階として，ここでは研究報告や修士論文など，学術論文に準じるコンテンツ（以下，準論文）を判定対象に加える。

第 1 表は既存の検索サービスと本研究の目指す検索システムの違いをまとめたものである。この表で一般的な検索エンジンに関する項目をすべて“△”としているのは，一般的な検索エンジンは収集対象とする範囲が大きいため，検索結果中に学術情報が出力はされるが，他の一般的なサイトやページに埋もれることで利用者が実質的にはアクセスしにくいことを示している。

D.　本研究の目的と手順

今回の研究では，ウェブコンテンツ中の PDF ファイル群からの①学術論文のみを対象とした自

第 1 表　学術情報流通に関わる既存の検索サービスと本研究の目標

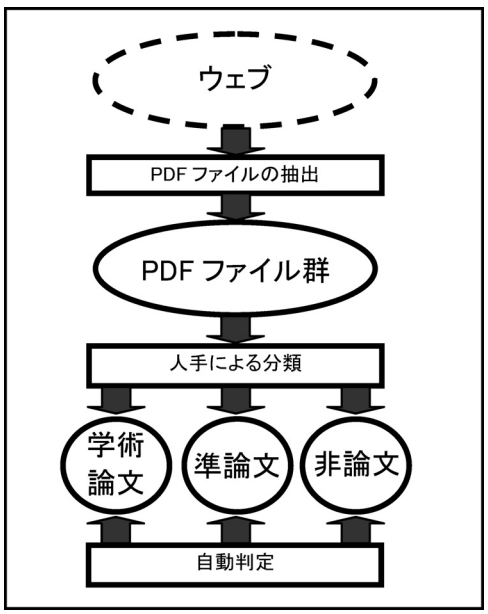
		商用 データベース	学術情報専門 検索エンジン	一般的な 検索エンジン	本研究の目標
オープンアクセスの実現方法	セルフアーカイビング	×	×	△	○
	機関リポジトリ	△	○	△	○
	オープンアクセス ジャーナル	△	○	△	○
備考				アクセスできるが 検索結果に埋没	論文に準じるコンテンツの自動判定

動判定，②準論文を含めた場合の自動判定，という二つの課題について判定実験を行う。具体的な手順としては第 2 図のように，①インターネット上で公開されている PDF ファイルを収集し，②そこからサンプルとなる PDF ファイルを無作為に抽出し，③人手による学術論文/準論文の判定から正解集合を作成し，④複数の判定手法を用いて自動判定を行い，⑤正解集合と判定結果から精度・再現率・*F* 値により評価を行う。

II. 実験集合の作成

A. PDF ファイルの収集

PDF ファイル集合の作成については，2005 年 5 月と半年後の 2005 年 11 月との二度にわたって行った。まず，ipadic2.5.1 の六つの名詞辞書ファイル（計 213,020 語）から，それぞれ，9,750 語（第 1 回目），10,250 語（第 2 回目）を無作為抽出し，各々の語について，サーチエンジン (Yahoo! Japan) を用いて検索を行った。その際，検索対象を「PDF ファイル」かつ「日本語」に限定するとともに，各検索語の最大収集件数は上位 100 件までとした。出力結果の重複除去後の異なり URL 件数は，それぞれ，307,514 件（第 1 回目）と 441,598 件（第 2 回目）となった。次に，各々の URL に対して PDF ファイルのダウンロードを試みた。ダウンロードが不可能であったもの，及び，0 バイト・ファイル，破損ファイル，暗号化ファイル，拡張子が “pdf” であるにもかかわらず PDF ファイルでないもの等を除去した結果，第 1 回収集では 248,314 件，第 2 回収集では 349,971 件の PDF ファイル集合が得られた。さらに，第 1 回目と第 2 回目の重複を除去したところ合計で 599,673 件となった。



第 2 図　判定実験の手順

B. 学術論文と準論文の判定

全体の PDF ファイル集合から，20,000 件を無作為に抽出し，6 人の判定者が各 PDF ファイルについて学術論文，準論文，非論文のいずれであるかを判定した。12,000 件を判定した時点で，学術論文と準論文と判定された 565 件のファイルを改めて 6 人全員が再判定し，判定基準の統一を図った。その後，残りのファイルについて修正された判定基準を適用した判定を行い，できるかぎり判定に揺れがないようにした。

学術論文の判定規準として，(1) 論文の形態をとっている，(2) タイトル，著者名，所属機関が明記されている，(3) 引用，参考文献がある，(4) 1 論文が 1 ファイルで構成されている，(5) 2 ページ以上である，を用いた。準論文の基準は学術論文の判定基準に一部満たないもの，内部向けのインフォーマルな研究報告を含めた。具体的には「研究ノート」「(学術雑誌以外での) 研究報告」「口頭発表原稿」「複数の論文の集合体」「論文・学術書の断片」「卒業研究・修士論文」「授業教材」などである。

なお，本研究では日本語の PDF ファイルを対象にしているが，収集手段として検索エンジンを用いたため，検索エンジンによる言語判定が誤ったと考えられる外国語（特に中国語）のファイルも一部に含まれていた。これらは非論文と判定した。

C. 実験集合の特性

1. 実験集合の基本的な属性

学術論文と論文以外（以下,「非論文」と示す）のファイル数，ファイルサイズ，ページ数，ページのレイアウトが縦長の割合を第 2 表に示す。第 2 表から，PDF ファイル集合 20,000 件中の論文数は 326 件と少なく，準論文を含めても 950 件であり，割合は 4.75% と 5% に満たないことがわかる。この点で自動判定を行う属性である論文か否かに関しては，非常に偏った集合であるといえる。

平均ファイルサイズは PDF ファイルのファイルサイズをバイトで示したものであり，論文，準論文の方が非論文よりも大きいことがわかる。平均ページ数は論文よりも準論文と判定されたものの方が多く，非論文は少ないページ数のものが多いことがわかる。

この表で縦型の割合は PDF ファイルに含まれるページレイアウト情報から 1 ページ目のページの縦の長さと横の長さを取得し，比較したときに，縦長であったファイルの割合である。論文ではすべてのファイルのレイアウトが縦長であり，準論文，非論文の順に徐々にその割合が低くなっていることがわかる。準論文と判定された横長レイアウトのファイルの多くは，学会発表や授業教材のスライド資料を PDF 化したファイルである。

2. jp ドメインにおけるサブドメインの分布

第 3 表は収集された URL のトップレベルドメインで大勢を占めた jp ドメインにおけるサブドメインの上位 5 位まであげたものである。第 3 表から，学術論文と準論文のサブドメインは ac.jp が多いことがわかる。一方，非論文のドメイン，サブドメインは多種多様であり，jp ドメインにおいても過半数を占めるような支配的なサブドメインはないが，中では co.jp が多い。非論文に関してはその他が多いが，これは地域型ドメイン，汎用ドメインが大多数となっている。現在は jp ドメインにおける地域型ドメイン，汎用ドメインの登場によって，サブドメインのバリエーションが多岐にわたるようになってきている。しかし，論文が置かれるのは以前からサイトを持つ確固たる組織のことが多く，地域型ドメインや汎用ドメインは使われない。一方，非論文が置かれるのは一般的なサイトが多く，より新しいタイプのドメイ

第 2 表　実験集合の基本的な属性

	学術論文	準論文	非論文
件数	326	624	19,050
平均ファイルサイズ	497,622.7 bytes	436,736.4 bytes	295,111.9 bytes
平均ページ数	10.94 pages	13.86 pages	6.88 pages
レイアウトが縦長の割合	100.00%	98.54%	92.50%

第 3 表　JP サブドメインの分布

ドメイン	学術論文		準論文		非論文	
ac	172	52.60%	269	43.32%	1,749	9.30%
go	52	15.90%	109	17.55%	1,889	10.05%
co	29	8.87%	47	7.57%	3,023	16.08%
or	24	7.34%	59	9.50%	2,127	11.32%
ne	5	1.53%	20	3.22%	1,322	7.03%
その他	45	13.76%	117	18.84%	8,687	46.21%

ンを使う傾向があるためと考えられる。

このように論文と非論文のPDFファイルのURLにおけるドメイン，サブドメイン間にかなり明確な違いがあることから，ドメイン，サブドメインは論文の自動判定の有力な手がかりの一つとなると考えられる。

3. 論文の主題分野

論文あるいは準論文と判定された950ファイルについて，図書館・情報学における資料組織化の専門的知識を持つ判定者2名が分類を行った結果である，論文ファイルの分野の分布を第4表に示した。論文の分野には技術工学が多く，次いで自然科学，社会科学の順になっている。ウェブ上で公開されている論文に，自然科学，技術工学が多いのは容易に予想できるが，一方でそれらに匹敵するほどの人文社会科学分野の論文も公開されていることがわかる。自然科学分野と比較して，全文データベースの提供が遅れている人文社会科学分野でもウェブ上での研究成果の公表は積極的に行われていると考えられる。

D. テキスト抽出とトークン化

1. PDFファイルからのテキスト抽出

判定実験に用いる判定器は，PDFファイルを直接扱うことはできないため，PDFファイルからテキストデータを抽出する必要がある。PDFファイルからテキストデータを抽出するための手法としては，①Adobe Acrobat[13]，②Xpdf 3.01pl2[14]，③PDFDocText[15]，④PDFTrans[16]といったプログラムを用いることが考えられる。しかし，実際に日本語PDFファイルを変換したところ，①Acrobatでは自動化することが困難であり，③PDFDocTextでは日本語以外の箇所で空白が除去される，一定の文字数で改行が入る，④PDFTransでは正しく日本語を扱うことができないなどの問題があることが明らかとなった。そこで，本実験では，Xpdfを用いて，PDFファイルからテキストデータの抽出を行った。

第4表 論文の主題分野

NDC	論文		準論文	
00 総記	33	10.1%	22	3.5%
10 哲学	20	6.1%	15	2.4%
20 歴史	12	3.7%	22	3.5%
30 社会科学	64	19.6%	161	25.8%
40 自然科学	64	19.6%	175	28.0%
50 技術/工学	88	27.0%	145	23.2%
60 産業	22	6.7%	60	9.6%
70 芸術/美術	4	1.2%	14	2.2%
80 言語	10	3.1%	5	0.8%
90 文学	9	2.8%	5	0.8%

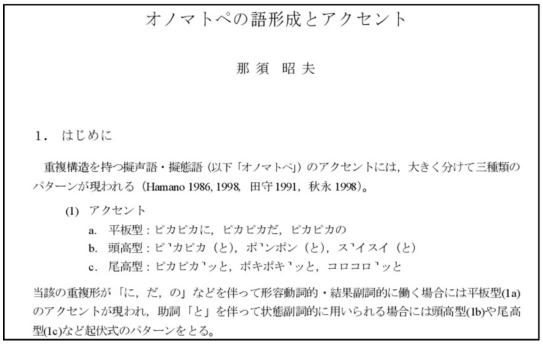
PDFファイルは表示・印刷時にレイアウトが再現可能なデータ形式であり，内部的には文書構造の情報をも保持することが可能である。しかしながら，多くのPDFファイルは単にレイアウト情報しか持たない。そのため，テキストデータの抽出を行うと，Xpdfはレイアウトの指定がされている箇所を改行・空白へと変換することが多い。

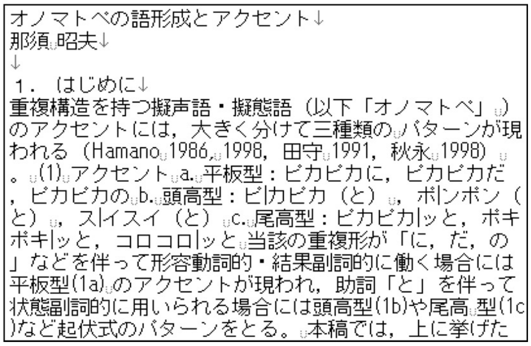
ここで，実際にPDFファイルからXpdfによってテキストデータを抽出した例を第3a図，及び，第3b図に示す。第3a図のように正しく変換できた場合には元のPDFファイルをほぼ正確に再現できていることがわかる。ただし，第3a図中では記号で改行部分示されているが，これが一部欠落し，また，一部の空白がレイアウトを再現するために追加されていることがわかる。

一方で第3b図は二段組のPDFファイルからのテキストの抽出例であるが，段組解除がきちんと行われていないため，段組の左部分から始まった文が空白を挟みそのまま段組の右部分につながっていることがわかる。具体的には，第3b図左で「はじめに」の1行目最後から2行目最初にかけて「アムステルダム条約」という語があるが，段組解除の失敗により，第3b図右では「アムス」と「テルダム条約」の間に段組右の行が挿入されている。

PDFファイルからのテキスト抽出が正しくできない主な原因は，段組をはじめとするレイアウ

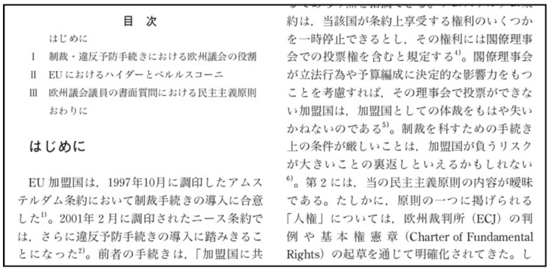
元 PDF ファイル　　抽出テキスト(改行、空白を記号で示す)

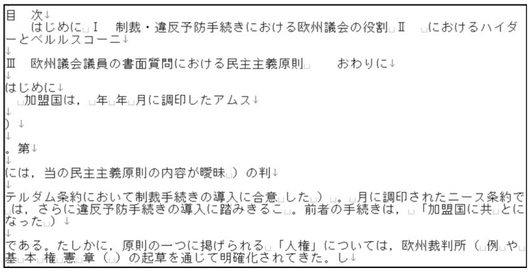




第 3a 図　変換が正しくできた例
出典：那須昭夫．オノマトペの語形成とアクセント．日本語・日本文化研究，no. 11, 2001, p. 9 上段

元 PDF ファイル　　抽出テキスト(改行、空白を記号で示す)





第 3b 図　変換が正しく行われなかった例
出典：山本直．EU の対加盟国制裁権限 - 欧州議会および欧州政党の対応を中心にして．阪南論集社会科学編，vol. 39, no. 2, p. 63 下段

トの指定，数式，図表，特殊文字，特殊なフォント指定であった。特に，レイアウトの指定はテキスト抽出を行った際に一番多く出現し，扱いが困難な問題となる。具体的には，元の PDF ファイルの作者による意図された改行・空白と，レイアウトの指定が Xpdf によって変換された改行・空白を判別することがほぼ不可能である，という問題となる。

そのため出現語を属性として用いた場合，①改行・空白の除去等の特別な後処理は行わない，②改行・空白は英数字の前後では空白 1 文字に変換し，英数字の前後以外の場合には除去する，という異なる二つのテキスト処理手法を施した実験集合群を作成した。②において改行・空白処理を英数字の前後とそれ以外で分けたのは，英単語の連結を防ぐためである。以下において，判定実験の対象が①の場合には「改行・空白処理なし」，②の場合には「改行・空白処理あり」として言及する。

2. トークン化

SVM などの機械学習手法で，出現語を属性として用いて，論文の自動判定を行う場合，日本語は膠着語であるため，テキストデータを，トークン（文字列や単語）に分割していく必要がある。トークン化には，形態素解析システム MeCab 0.81[17] と単純にテキストデータを 2 文字ごとに分割した bigram を用いた。以下では前者によって形態素に分割したものは mecab, bigram によって分割したものは bigram として参照する。なお，切り出したトークンからの選択は行わず，すべてのトークンを自動判定の手がかりとして用

いた。

III. 実験環境

A. 判定に用いた属性

1. 出現語アプローチ

学術論文の判定はテキスト分類の課題の一つであり，まずテキストの内容に関する属性として，PDF ファイル中に出現する語を手がかりとすることを考えた。この属性を用いた判定を後述のルールベースアプローチと区別するため，以下では出現語アプローチと表記していく。

この出現語によるアプローチは，従来の研究成果も多く，実績のあるテキスト分類の判定器を用いることができる。ただし，出現語アプローチでは，属性数が比較的少ない mecab でも 77,814 件となり，属性の選択等を行わない場合，応用可能な判定器は限定されてしまう。

しかしながら，あえて属性の選択を行わなかった理由の一つは，今回の実験では論文の自動判定の対象データについて分野を限定せずに収集したことがある。このような判定においては，専門分野に固有の専門用語，主題語といった内容語ではなく，機能語に論文特有の文体が現れることを期待するが，何らかの手法で属性の選択を行った場合に，機能語に関する情報が欠落することを避けたのである。

2. ルールベースアプローチ

筆者らによる先行研究では，ウェブコンテンツからの論文の自動判定は膨大な非論文中から非常に限られた数の論文を抽出する難度の高い判定行為であり，出現語だけからのアプローチだけでは不十分なことが示唆された[18]。そこで，出現語によるアプローチだけでなく，他の属性を用いたアプローチも行った。

現時点では人があるファイル群から論文を論文として判断する行為に対する体系的な研究がないため，自動判定の手がかりとなる属性群を決定することはできない。しかし，人があるファイルを論文と判断する際には，論文の内容だけでなく，論文のレイアウト，構造的な特性，入手元等のさまざまな要素を総合的に用いるはずである。ここでは，経験的に論文と関係すると考えられること，PDF ファイルから入手可能かつ判定器に投入可能なことの 2 点を条件として，以下の第 5 表に示した 4 カテゴリ，19 の属性（ルール）を採用した。第 5 表における出現キーワードとは経験的に論文中に出現するだろうキーワード群を挙げたものであり，行ごとに類義語，同義語をまとめている。例えば下から 2 行目の「引用文献」「参考文献」「参照文献」に関しては，ファイル中にこれらのいずれかの語が出現すれば，この属性は「出現した」ものとし，自動判定の手がかりとする。

このような属性を用いるためにはあらかじめ学術論文に関する知識が必要であるため，汎用性には欠けるが，出現語アプローチと比較して，非常

第 5 表 ルールベースの判定で用いた属性

カテゴリ	属性
構造	ファイルサイズ
	ページ数
	ページの形（縦型か横型か）
入手元	URL が ac.jp であるか
	URL が go.jp であるか
文体	文末が「である」調か「ですます」調か
	会話が出てくるか （文末に「ね。」,「 」が使われているか）
	ひらがなが出現するか（外国語か）
出現キーワード	「研究」
	「文献」
	「被験者」
	「調査」「分析」「実験」
	「紀要」「研究報告」「研究ノート」
	「図」「表」
	「本稿」「本研究」「本論文」
	「研究成果」「研究結果」
	「考察」「考慮」
	「引用文献」「参照文献」「参考文献」
	「大学」「研究所」「研究センター」

に少ない属性からの判定となるため，限られた機械資源や時間という点からはより有利なアプローチとなる。

B. 判定手法とその実装

判定性能の向上を図り，各判定手法の特性を比較するため，採用する判定手法を可能な限り幅広い観点から検討した。

出現語アプローチでは，テキスト分類において評価の高い， SVM, AdaBoost, そして， スパムフィルタとして広く使われているベイジアンフィルタの 3 種類の判定手法として用いた。ルールベースアプローチでは， SVM, AdaBoost に加えて，ナイーブベイズ，決定木 (C4.5), メタ判定手法として Vote からの判定を行った。

各判定手法を実装したシステムとして，ルールベースアプローチでは， Weka (Waikato Environment for Knowledge Analysis) を用いた。Weka[19) 20)] は Waikato 大学（ニュージーランド）の機械学習センターを中心に Java 言語で開発が行われているデータマイニングツールであり，数多くの機械学習に基づく判定器を実装している。原則的には Weka3.4.7，必要な場面では開発版である Weka3.5.2 を用いた。Weka は出現語によるアプローチでは使用しなかった。これは，前述のように，実験集合における出現語は高次元の属性群であり，Weka では扱うことができなかったためである。そのため，出現語によるアプローチでは各判定手法について異なる実装を用いている。

1. サポートベクターマシン

サポートベクターマシン（以下， SVM）は，Vapnik によって提案された 2 クラス分類器の一種である[21)]。SVM は正の例と負の例を分離する平面を構成し，その分離平面に最も近い例（サポートベクター）同士のマージン（サポートベクターと分離平面の最小距離）を最大化することで学習が行われる。これをカーネル関数により高次元空間に写像することで，高次元空間においても線形分離を行うものである。

SVM は高い汎化性能を持ち，カーネル法により非常に高次元のデータを扱うことができる点が特徴である。投入する属性数が多くなりがちなテキスト分類においても多くの応用事例があり，後述の AdaBoost ともにテキスト分類では他の手法と比べ高い性能を示している。

SVM の実装としては，出現語アプローチでは，SVM[light] 6.01[22)] を用い， カーネル関数には線形カーネル関数を用いた。ルールベースでは，Weka から外部の LIBSVM 2.81[23)] を呼び出す形で用い，出現語アプローチと同様にカーネル関数は線形カーネル関数とした。

2. AdaBoost

ブースティング (Boosting) 法は，バッギング (Bagging) と同様に複数の判定器を組み合わせる，集団学習 (ensemble learning) と呼ばれる枠組みを用いた機械学習手法の一つである。複数の判定器（弱学習器）の組み合わせ方，重み付けを学習することで単一の判定器を用いる場合に比べ性能を改善しようとするものである。

AdaBoost は初期のブースティング法を改良したもので，比較的単純な判定器を用いるため，計算量が少ない，過学習が起こりにくいという特徴を持つ。Schapire と Singer による実験では，単語の有無による弱学習器を AdaBoost によって組み合わせた判定器が最近傍法（k-NN 法）やナイーブベイズ法による判定器よりも高い判定性能を示している[24)]。

ただし， AdaBoost は学習集合において判定が易しい事例よりも難しい事例を集中的に学習することで精度を高めていく。このため，例外的な事例が多い場合，性能悪化を招くことが指摘されている[25)]。

SVM との関係ではマージンの理論に基づく点では非常に似通っているが，「異なるノルムは異なるマージンに対応しうる」「必要な計算量が違う」「高次元での探索を効率的に行うために異なるアプローチを用いている」点が異なっている[26)]。

今回は AdaBoost の実装として，出現語に関し

ては，BoosTexter の AdaBoost.MH を用いた。BoosTexter[27] を用いた理由は，出現語は mecab で分割した場合，70 万以上と次元数が多いが，次元縮約なしに扱うことができる AdaBoost 実装であるからである。BoosTexter に実装された AdaBoost 実装は 3 種類であるが，AdaBoost.MH が先行研究で他の 2 種類に優る結果を出しているため[28]，AdaBoost.MH を用いた。ルールベースでは Weka のモジュールを用いた。両方ともに AdaBoost を弱学習器として決定木アルゴリズムの一種である決定株 (decision stumps) と組み合わせ，学習の繰返しラウンド数を 10 回，100 回，1000 回としたときの判定を行った。ここで決定株とは単一ノードから構成される非常に単純な決定木である。

Weka では決定株以外の判定器も弱学習器として組み合わせることが可能であるが，出現語アプローチで用いた BoosTexter では決定株しか弱学習器として実装されていないため，ここではルールベースアプローチにおいても，AdaBoost と組み合わせる弱学習器は決定株とした。

3. ナイーブベイズ/ベイジアンフィルタ

ナイーブベイズ (naive Bayesian classifier) は，ベイズの確率モデルに基づく，単純な分類器である。“naive” とは各属性同士が独立である，潜在的な属性が影響しないという仮定から名づけられたものであるが，単純であるがゆえに理論的な拡張が容易であり，応用範囲も広い。

ベイジアンフィルタ (Bayesian Filter) は，ナイーブベイズを応用したものであり，現在では，主として電子メールの中からスパムメールを検出するシステムで用いられている。特に Paul Graham による “A plan for spam”[29] が発表されて以降，多くのシステムが開発されている。ベイジアンフィルタをスパムメールに応用する場合，非スパムメールとスパムメールに出現するトークンに対するスパム確率を学習し，そのスパム確率をもとに，新たに受信した電子メールに対して，スパムメールの判定を行う。スパムメールは，内容からも判定することは可能であるが，内容だけでなく，件名の書き方などのスタイルが判定に有効であるといわれている。

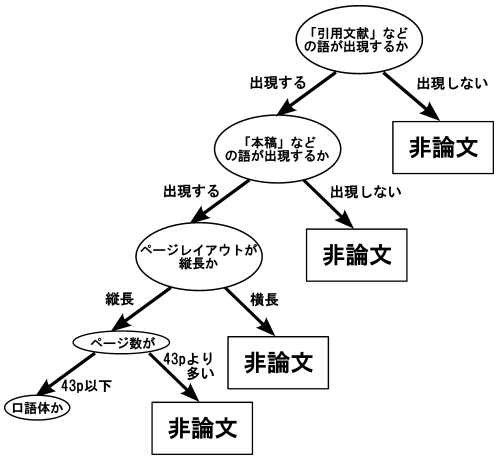
出現語アプローチにおけるベイジアンフィルタの実装としては，日本語にも対応可能である bsfilter[30] を用いた。bsfilter には有名な Paul Graham 方式も実装されているが，より精度が高いとされる Gary Robinson–Fisher 方式[31] を用いた。bsfilter は，各ファイルに対してスパム確率を算出する。スパムメール判定に用いる場合には，この確率が高いとスパムメールであると判定されるが，本実験では，「非論文」として判定する。ルールベースアプローチでは，Weka のナイーブベイズの実装であるモジュール，NaïveBayes を用いた。

4. 決定木 (C4.5)

決定木 (decision tree) は属性条件により分岐するノードから構成される木構造を用いた伝統的な機械学習手法の一つである。代表的な決定木アルゴリズムとしては，CART[32], ID3[33], C4.5[34] があるが，ここでは Weka に実装されている C4.5（モジュール名は J48）を学習結果の分析のために用いた。

決定木アルゴリズムの特徴は，与えられた属性に関する if-then ルールで木が構築されるため，他の手法と比べて可読性の高いことである。実際に交差検定用一番目の集合の判定に使われた決定木の一部を第 4 図に挙げた。この図からは構築された決定木からどのような学習がされたか，どのような判定がなされるかを容易に理解することが可能である。例えば，図において，最上部のノードは引用文献があるかどうかを示しており，引用文献がなければ非論文とし，引用文献あれば次のノードで「本稿」などの語が出てくるかを判定し，出てこなければ非論文とする判定が行われていることを示している。

また，決定木は，近年では AdaBoost などの集団学習の弱学習器として使われることも多い。今回の実験では，AdaBoost の弱学習器としては非常に単純な決定木である決定株と組み合わせている。



第 4 図　生成された決定木の一部

5.　メタ判定器 (Vote)

ルールベースの判定では複数の判定器を組み合わせたメタ判定器も用いた。これは，学術論文の判定という難しい課題に対して，できるだけ多くの観点からの予測を用い判定性能向上を目指すという積極的な理由がある。それ以外に，ルールベースの判定では属性数が少なく，判定に必要な機械的資源に対する負荷が低いため，複数の判定器を同時に用いることが可能であること，計算時間が短いため複数の判定器で判定を行っても実用的な速度での判定が可能であること，という理由もある。

実験では Weka の Vote モジュールを用い，論文判定に失敗した SVM を除くナイーブベイズ，AdaBoost (100), 決定木の三つの判定器の予測値を組み合わせ行った。

C.　評価尺度

この研究では精度 (P), 再現率 (R), F 値（F_1 値，F_2 値）を評価のために用いた。

精度はどれだけ正確に判定できたかを，再現率はどれだけ網羅的に判定できたかを示す。ただし，原則的に精度と再現率は反比例の関係にあるため，精度だけあるいは再現率だけから評価することはできない。そこで，総合的な指標として F 値を用いた。F 値は α の値によって，精度と再現率の重みを変えるが，$\alpha=0.5$ とした場合，つまり精度と再現率の調和平均の値としたものが一般的には用いられる。しかしながら，このシステムの目的である論文あるいはそれに準じるコンテンツの自動判定を想定した場合，再現率がより重視されると考えられる。そのため，ここでは $\alpha=0.5$ の場合の F_1 値，再現率をより重視した $\alpha=1/3$ とした F_2 値の両方を比較に用いている。

$$P=\frac{\text{システムが判定した正解件数}}{\text{システムが論文と判定した件数}}$$

$$R=\frac{\text{システムが判定した正解件数}}{\text{全論文件数}}$$

$$F=\frac{1}{\alpha\cdot 1/P+(1-\alpha)\cdot 1/R}$$

本実験では，学習用・判定用データを分割し，4 交差検定を行ったが，各データセットにおいて，各評価尺度の値を求め，それらを平均した値を算出した (macro-averaging)。

IV.　自動判定実験の結果

自動判定実験は，本研究の二つの目的に対応した形で，人手により学術論文とされたもののみを論文として判定した場合と，論文だけでなく準論文と判定されたものも含めて論文とした場合について行った。

なお，実験結果を第 6〜11 表に示したが，数多くの手法の組み合わせに関して実験を行ったため，これらの表では評価尺度ごとの最高値を太字で強調してある。

A.　学術論文を対象とした判定

今回の実験では，判定の手がかりとなる属性で 2 種類，出現語アプローチでは判定手法 3 手法，空白処理あり / なし，トークン化では mecab / bigram，ルールベースアプローチでは判定手法 4 手法，さらに AdaBoost ではラウンド数により 3 種類と数多くの組み合わせにおける値を出しているため，全体を一度に提示すると比較が困難となる。そこで以下では各アプローチ別に各組み合わせに関する比較を行い，最後に性能的に特徴が出ている代表的なもの同士を取り上げ，分析を行っ

ている。

1. 出現語アプローチの結果

学術論文のみを対象として出現語を手がかりとした自動判定の結果を第 6 表に示した。

改行・空白処理を行った場合と処理しない場合を比較すると，全体的に改行・空白処理を行った場合の値は行わないときよりも向上している。しかしながら，各手法間での値の差と比べた場合，その違いは大きいものではなく，明確な傾向は見られなかった。

次に手法ごとに比較すると，精度で最も高い値を出したのは，SVM において空白を除去し mecab でトークン化した場合である。SVM は他の組み合わせにおいても 7 割を超える高い精度を出している。再現率が高い手法は改行・空白処理を行わない bigram でトークン化したベイジアンフィルタによる判定であり，値自体は .933 と 9 割を超えているが，一方で精度は .026 と低い値である。第 6 表において「判定論文数」は自動判定において学術論文と判定された数の平均を示しているが，ベイジアンフィルタの行を見ると判定論分数の数が他と比較して非常に多いことがわかる。これはほとんどのデータを論文と判定した結果であるため，再現率の値は高いが意味がある判定とは言い難い。総合的な F_1, F_2 値が高いのは AdaBoost である。AdaBoost ではラウンド数が 100 回から 1000 回に増えたときに，精度が高くなる一方で再現率が下がっている。

2. ルールベースアプローチの結果

第 7 表はルールベースアプローチにおける学術論文に関する自動判定の結果を示したものである。なお，SVM はすべて非論文と判定し，論文と判定した件数は 0 件であった。このため，精度，F 値の値は算出不能であり，第 7 表では “N/A” としている。

評価尺度別に比較すると，精度に関しては AdaBoost のラウンド数が 1000 回のときが最も高い値となっている。一方で再現率ではナイーブベイズが 9 割に近い値を出している。総合的な尺度である F 値に関しては F_1, F_2 値共に Vote が高

第 6 表　出現語アプローチによる論文の自動判定

改行・空白	手法		トークン	精度	再現率	F_1 値	F_2 値	判定論文数
処理あり	SVM		mecab	**.750**	.277	.404	.350	30 件
			bigram	.727	.274	.398	.346	31 件
	AdaBoost	Round 10	mecab	.521	.403	.455	.436	63 件
		Round 100	mecab	.549	.407	.467	.445	60 件
		Round 1000	mecab	.605	.383	**.469**	.437	52 件
	ベイジアンフィルタ		mecab	.039	.914	.076	.109	1,891 件
			bigram	.047	.923	.090	.128	1,598 件
処理なし	SVM		mecab	.713	.273	.395	.344	31 件
			bigram	.743	.271	.397	.344	30 件
	AdaBoost	Round 10	mecab	.413	.367	.389	.381	72 件
		Round 100	mecab	.527	.417	.465	**.448**	64 件
		Round 1000	mecab	.547	.387	.453	.428	58 件
	ベイジアンフィルタ		mecab	.103	.506	.172	.220	399 件
			bigram	.026	**.933**	.050	.074	2,937 件

くなっている。系統の異なる手法を組み合わせることで各手法の利点が出た結果といえる。Vote は単純な組み合わせであるため，高度な組み合わせを行う手法を用いることで一層の最適化が可能になるとも考えられる。

この結果において特徴的なのは，SVM が 1 件も論文を正しく判定できなかったことである。原因としては，属性の選択に問題があったとも考えられる。しかし，一般的には SVM よりも性能が低いとされる他の判定手法ではある程度の正解が得られたことを考慮すると，SVM が完全に判定に失敗したことは，今回の自動判定の難度の高さを示唆している。

なお，決定木 (C4.5) アルゴリズムによって作成された決定木の例として，交差検定用一番目の集合について前述の第 4 図にあげたが，他の交差検定用実験集合でも最上位のノードは「引用文献」に関する語が出現するかであった。ここから，引用文献があるかが論文の判定の重要な手がかりの一つであることがわかる。

3. 全体的な結果

学術論文を対象とした自動判定に関して，出現語アプローチとルールベースアプローチにおける代表的な判定結果を第 8 表に示した。各評価尺度において最も高い手法群を選択にしているにもかかわらず，この表において精度・再現率の両方が同時に 50% を超える手法がない点からは全体的に十分な判定性能が得られたとはいえない。

精度・再現率の点からは，出現語アプローチの SVM の場合には 75%以上の高い精度で論文を検出できていた。また，.933 と再現率の値が高いのはベイジアンフィルタであるが，この手法は .026 と精度の値が絶対的に低くなっている。精度とのバランスからは，ルールベースのナイーブベイズがある程度の精度は確保しつつ，再現率の値

第 7 表　ルールベースアプローチによる論文の自動判定

手法		精度	再現率	F_1 値	F_2 値	判定論文数
ナイーブベイズ		.233	**.893**	.370	.459	312 件
決定木 (C4.5)		.430	.236	.305	.278	45 件
AdaBoost	Round 10	.422	.331	.371	.357	64 件
	Round 100	.467	.393	.427	.415	69 件
	Round 1000	**.504**	.365	.423	.402	59 件
SVM		N/A	.000	N/A	N/A	N/A
Vote		.444	.537	**.486**	**.502**	99 件

第 8 表　学術論文に関する自動判定

属性	手法	トークン	空白改行	精度	再現率	F_1 値	F_2 値	判定論文数
出現語	SVM	mecab	処理あり	**.750**	.277	.404	.350	30 件
	AdaBoost (R100)	mecab	処理なし	.527	.417	.465	.448	64 件
	AdaBoost (R1000)	mecab	処理あり	.605	.383	.469	.437	52 件
	ベイジアンフィルタ	bigram	処理なし	.026	**.933**	.050	.074	2,937 件
ルール	ナイーブベイズ			.233	.893	.370	.459	312 件
	AdaBoost (R1000)			.504	.365	.423	.402	59 件
	Vote			.444	.537	**.486**	**.502**	99 件

が高い手法といえる。メタ判定の Vote を除き精度よりも再現率の値が高かったのは，ベイジアンフィルタとナイーブベイズによる判定だけである。

F 値からみると，メタ判定の Vote が最も高い値を示したが，それを除けば，F_1 値では，出現語アプローチの AdaBoost (1000) の mecab が .469 と高い値を示した。再現率重視の F_2 値ではナイーブベイズの値が高い結果となった。

ルールベースアプローチは用いる属性が 19 と少ないにもかかわらず，F 値に関して出現語アプローチと比較したときに，ほぼ同様か，より高い値を出した。

B. 学術論文と準論文を対象とした判定

ここでは人手で学術論文と判定されたものだけでなく，準論文と判定されたものも正解として，学習，自動判定を行った場合の判定実験の結果を示す。

学術論文のみを対象とした実験と同様に各手法に関する組み合わせの種類数が多いため，最初に出現語アプローチとルールベースアプローチごとの結果を比較し，全体的な分析は各評価尺度の値の高い代表的な手法についてのみ行う。

1. 出現語アプローチの結果

出現語アプローチにおける準論文も含めた場合の自動判定の結果を第 9 表に示した。全体的には，出現語アプローチの学術論文のみを対象とした判定と比較して，精度，再現率の値が 10〜20% 程度高くなっている。

改行・空白処理を行った場合と処理しない場合を比較すると，学術論文のみを対象としたときと同様に，全体的に改行・空白処理を行った場合の値は行わないときよりも向上している。しかし，やはり，違いは大きいものではなく，明確な傾向は見られなかった。

評価尺度別に値の良かった手法としては，精度の値が高かったのは改行・空白処理を行った SVM の bigram であり，学術論文のみを対象としたときと同様に SVM ではあるが，トークン化の手法が mecab ではなく bigram の場合の値が

第 9 表　出現語アプローチにおける準論文を含めた自動判定

改行・空白	手法		トークン	精度	再現率	F_1 値	F_2 値	判定論文数
処理あり	SVM		mecab	.742	.482	.584	.546	154 件
			bigram	**.749**	.478	**.584**	.544	152 件
	AdaBoost	Round 10	mecab	.580	.432	.495	.472	177 件
		Round 100	mecab	.624	.515	.564	.547	196 件
		Round 1000	mecab	.675	.513	.583	**.557**	180 件
	ベイジアンフィルタ		mecab	.113	.895	.200	.270	1,889 件
			bigram	.136	.913	.236	.314	1,597 件
処理なし	SVM		mecab	.740	.469	.574	.534	151 件
			bigram	.736	.470	.574	.535	152 件
	AdaBoost	Round 10	mecab	.587	.379	.461	.430	154 件
		Round 100	mecab	.622	.478	.540	.518	182 件
		Round 1000	mecab	.633	.500	.559	.538	188 件
	ベイジアンフィルタ		mecab	.257	.432	.322	.352	399 件
			bigram	.075	**.931**	.139	.194	2,937 件

高くなる結果となった。再現率の値が良かった手法は改行空白処理を行わないベイジアンフィルタの bigram であったが，これは学術論文のみの場合と同様であり，さらにベイジアンフィルタの精度の値は .026 から .075 と向上している。しかし，多くの非論文を論文と判定した結果の値であり，絶対的には低いと言える。F_1 値が高かったのは，改行・空白処理を行った SVM の bigram であり，F_2 値が高かったのは改行・空白処理を行った AdaBoost のラウンド数 1000 回のときであった。

AdaBoost は学術論文のみを対象としたときと同様に，ラウンド数が上がるほど精度が高くなる傾向が見られた。

2. ルールベースアプローチの結果

ルールベースアプローチにおける準論文も含めた場合の自動判定の結果を第 10 表に示した。出現語アプローチと同様に学術論文のみを対象とした判定と比較して，精度，再現率の値が 10〜20% 程度高い傾向となった。

評価尺度別に値の良かった手法としては，精度が高かったのは SVM が .681 と 7 割に近い値を出している。SVM は学術論文のみを正解とした実験ではすべてを非論文と判定してしまい値がでなかったことから，正解に準論文を含めた方が自動判定の難度がより低くなると考えられる。再現率の値が高かったのは学術論文を対象としたときと同様にナイーブベイズであるが，再現率の値は .893 から .726 と下がっている。精度は 10% 以上改善することから，準論文を含めることでより判定しやすい正解が増える一方で，正解集合の特性が弱まったためと考えられる。F 値に関しては，学術論文を対象としたときと同様にメタ判定器である Vote が最もよい値を出している。

3. 全体的な結果

第 11 表は，学術論文と準論文を対象とした自動判定で，出現語アプローチとルールベースアプローチにおける代表的な手法の結果を示したもの

第 10 表　ルールベースアプローチにおける準論文を含めた自動判定

手法		精度	再現率	F_1 値	F_2 値	判定論文数
ナイーブベイズ		.363	**.726**	.484	.545	475 件
決定木 (C4.5)		.662	.445	.532	.500	160 件
AdaBoost	Round 10	.642	.433	.517	.486	160 件
	Round 100	.654	.437	.524	.491	159 件
	Round 1000	.652	.425	.515	.481	155 件
SVM		**.681**	.436	.532	.495	152 件
Vote		.592	.551	**.571**	**.564**	221 件

第 11 表　準論文も含めた自動判定

属性	手法	トークン	空白改行	精度	再現率	F_1 値	F_2 値	判定論文数
出現語	SVM	bigram	処理あり	**.749**	.478	**.584**	.544	152 件
	AdaBoost (R1000)	mecab	処理あり	.675	.513	.583	.557	180 件
	ベイジアンフィルタ	bigram	処理なし	.075	**.931**	.139	.194	2937 件
ルール	ナイーブベイズ			.363	.726	.484	.545	475 件
	AdaBoost (R1000)			.681	.436	.532	.495	152 件
	Vote			.592	.551	.571	**.564**	221 件

である。自動判定において区別がつきにくいと考えられる準論文を含めることで，学術論文のみを対象とした場合よりも判定性能は全体的に向上している。

評価尺度別にみていくと，精度は出現語アプローチにおける改行・空白処理を行った bigram でトークン化した SVM であり，精度の値は学術論文のみを対象とした場合の最高精度を示した組み合わせと同様であるが，再現率が .277 から .478 と倍近く向上している。再現率の値は，出現語アプローチにおける改行・空白処理を行わない bigram のベイジアンフィルタが .931 と高いが，この組み合わせは精度の値が他の手法と比較して格段に低く，精度とのバランスからはルールベースのナイーブベイズが .363 の精度と .726 の再現率を両立させているといえる。F 値についてみると，精度と再現率のバランスをとった F_1 値は出現語アプローチにおける改行・空白処理を行った bigram でトークン化した SVM であり，再現率を重視した F_2 値ではメタ判定器である Vote が高い値を示す結果となった。

C. 誤り分析

今回の学術論文の自動判定実験に関して，今後の性能向上と判定手法の特徴をみるために，誤り分析を行った。ただし，実験では判定手法や属性などからの数多くの組み合わせによる判定を行っており，精度あるいは再現率が低い手法に関して誤り分析を行った場合，確認すべきデータは膨大な数になり，人手で精査することは難しくなる。そこで，今回の誤り分析の対象としたのは，交差検定のために四分割した一番目の集合を判定集合とし，残りの三集合を学習集合とした場合に，論文のみを正解として行った自動判定実験で，判定性能が高かった組み合わせにおいて誤った事例とした。

具体的には，最も精度が高かった出現語アプローチの改行・空白処理を行いトークン化に mecab を用いた SVM が「非論文を論文として誤って判定した例」と，再現率の高かったルールベースアプローチのナイーブベイズ法が「論文を非論文として誤って判定した例」を対象とした。誤り分析の対象とした例について，オリジナル PDF ファイルと変換されたテキストファイルを精査し，必要に応じて元 PDF ファイルの入手元サイトの周辺のページも確認した。

1. SVM の誤り分析

精度の値が高かった SVM が交差検定用一番目の集合に対して論文判定を誤った事例は 6 件であり，全てが人手では準論文として判定されたファイルであった。全てのファイルには参考/引用文献が付与されていた。

誤った事例 6 件の内訳は，紀要に掲載された論文的内容の文書が 1 例（内容は，ある言語の語彙表），体裁は論文とほぼ同様の研究報告書が 2 例，複数の研究論文の要旨が一つのファイルにまとめられていたものが 1 例，図書の一部が 1 例，講習会のテキストが 1 例であった。

上記のように，SVM が非論文を論文と誤った事例はすべて準論文と人手で判定されたものであり，明らかな非論文が含まれていなかったという点からは自動判定の難度が高い事例のみであったといえる。さらに，研究報告を論文と誤った事例が 2 例あることから，今後，SVM の判定の精度向上のために，学習集合に研究報告の事例を追加し，研究報告と論文の差異に注力して学習を行わせるといった方策が考えられる。

2. ナイーブベイズの誤り分析

論文の自動判定について再現率が高かったナイーブベイズにおいて交差検定用一番目の集合に対して非論文と誤って判定してしまった論文の事例は 6 件であった。

このうちテキスト抽出において失敗した事例が 1 件あった。これは，本文ではなく図表中のテキストのみが抽出されていたため，非論文と判定されたと考えられる。また，ページ数が 3 ページ以内の事例が 3 例あり，これらは参考文献・引用文献が少なく，人手による判定でも準論文との境界が曖昧な事例といえる。

残りの二つの事例に関して，一つは歴史研究で

あり，論文中に旧仮名遣いによる口語体文が多く引用されていた。そのため，ナイーブベイズの判定で非論文と判定されたと考えられる。もう一つの事例は，医学分野の動物実験に関するものであり，論文中に英語表記，さらにその短縮形の語が多く用いられていたこと，さらに二段組のレイアウトであったものの段組解除がされないままテキスト抽出がされてしまっていたため，多くの文章が完全な文として認識されていなかったことが原因として挙げられる。ただし，後者の点は他の正しく判定されている事例についても多く見られた問題であるため，それだけが原因とは考えにくい。

以上のことから，ナイーブベイズの再現率向上のためには，段組解除を含めてテキスト抽出の精度を上げること，論文中の会話部分を認識させることなどが考えられる。

V. 考察と今後の課題

A. 実験結果に関する考察

1. 学術論文のみの判定と準論文を含めた判定

今回収集された PDF ファイル集合から判断する限り，インターネット上から収集された PDF ファイルに含まれる学術論文の割合は 2% 程度と非常に低い。そのため，正解を学習論文のみとした場合，正誤の事例間の偏りが大きくなるが，このような偏りが大きい実験集合は機械的な自動判定は難しくなってしまう。実際に，今回の実験でも学術論文のみを対象とした自動判定実験では，精度・再現率の両方が共に 5 割を超える判定手法はなかった。

一方で学術論文ではないが有用な学術情報源となると思われる準論文を含めた自動判定では 10〜20% 程度の性能の向上が見られた。前述のように現在，準論文をも含めた形での学術情報専門の検索サービスがないことを考えたときに，このように公開された PDF ファイルすべてを対象とする自動判定では，準論文を含めた方がより現実的な解といえるかもしれない。ただし，学術論文のみを対象とするか準論文も対象に含めるかという問題は，後述のように学術論文の段階的な自動判定を行うことで，解消されるとも言える。

2. 出現語アプローチとルールベースアプローチ

学術論文の自動判定に関して，ルールベースアプローチでは 19 属性しか用いていないにもかかわらず，出現語アプローチに比べ遜色のない，あるいはそれ以上の性能を示した（第 8 表，第 11 表）。出現語アプローチと比較して，判定に用いる属性数が桁違いに少ないため，機械的資源に対する負担が少なく，判定処理にかかる時間も短いという利点がある。一方で，論文の判定に関する経験的な知識が必要であるため，他の問題領域への応用ができないという点で汎用性に欠けるという問題もある。

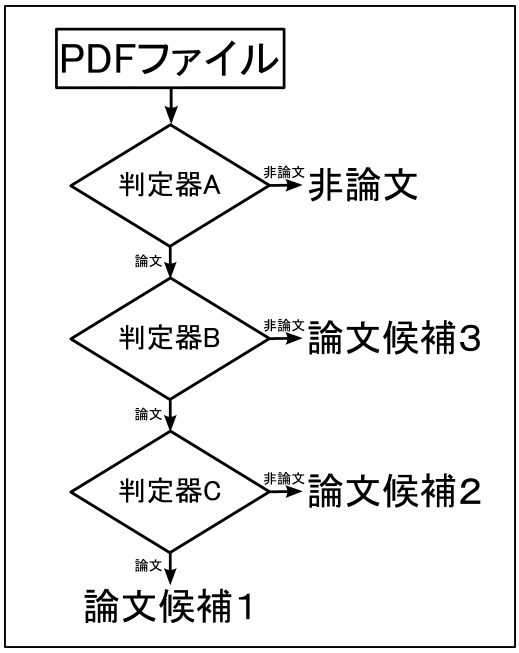
実際の判定システムにおいて，リアルタイムに複数の判定器を並列に用いる場合には，処理時間が短いルールベースアプローチによる判定手法が有利といえる。

3. 段階的な論文判定

学術情報を対象とした検索エンジンの実用化という観点からは，今回の実験では単独で精度と再現率の両方ともに十分な性能を示す論文の自動判定手法はなかった。この結果が，テキスト分類において性能が高いとされる SVM や AdaBoost を含め数多くの機械学習手法を応用したものであることを考慮すると，PDF ファイルからの論文自動判定は難度が高い処理であることがわかる。今後，各処理の段階における精緻化を行うことで，徐々に性能改善を図ることは可能と考えられるが，現時点で精度と再現率の値を両立させるような劇的な性能改善を行うことは難しいと予想される。

しかし，今回の実験から明らかとなった以下の 3 点を検討したときに，学術論文か否かの二値判定でなく，論文らしさについての段階的な自動判定を行うシステムであれば，現時点で十分に構築可能であると考えられる。

1) 精度，あるいは，再現率を単独で取り上げた場合には高い値を示す手法もあった。
2) 単純な重みなしの組み合わせにもかかわら



第5図 段階的な論文判定

ず，メタ判定器である Vote が総合的に高い性能を示した。

3) 準論文も含めた場合の自動判定は，少なくとも論文のみを対象とした場合よりもよい性能を示した。

具体的には，第5図のように判定器を再現率が高いものから精度が高いものまで直列に組み合わせ，各段階において学術論文と判定されたものを利用者に提示するというシステムである。

B. 今後の課題

今後，ウェブ上に公開されているコンテンツからの学術論文の自動判定を行っていく上で，今回の自動判定実験の結果からは，段階的な論文判定を実装した検索エンジンの構築，テキスト処理の一層の精緻化，属性のより適切な選択，といった課題を検討していく必要が示唆された。

1. 段階的な論文判定を実装した検索エンジンの構築

上記のように複数の判定手法を直列に連結し，論文の段階的な自動判定を実装した，学術論文の自動判定を組み込んだ検索システムを構築し，判定実験によりその有効性を検証することが考えられる。

今回の実験では精度を重視した場合には，出現語アプローチの SVM が，再現率を重視した場合にはルールベースアプローチのナイーブベイズが，単独の手法としてはある程度の性能を示した。そのため，実際に論文らしさの段階的判定を組み込んだ検索エンジンを構築する際には，ナイーブベイズで第一段階の判定を行うことで絞り込みを行い，ある程度，論文らしさを持ったデータに対して，SVM でより精度の高い判定を行っていく形を考えることができる。

2. テキスト変換処理の精緻化

今回の実験における改行・空白処理の有無に関する分析からも明らかなように，PDF ファイルからのテキスト抽出において数多くのノイズが混入してしまう。そのため，段組解除，変換ミス文字，空白，改行等の処理をより精緻化していくことが，論文の自動判定の性能向上につながると考えられる。

3. 属性群の選択

出現語アプローチでは，すべての出現語を用いた場合，属性数が膨大であるため利用可能な判定手法の実装に制限がでてくる。そこで潜在的意味インデキシング，主成分分析による次元縮約，あるいは，フィルタリングによる属性の選択を行うことで，応用可能な判定手法の選択肢を増やすことが可能となる。

また，ルールベースアプローチでは今回初めて論文の自動判定に有効と考えられる属性群を選定したが，今後，さらにより適切な属性を選定・追加していく作業も必要であろう。

謝　辞

本研究を行うにあたり，オープンアクセスに関する記述について慶應義塾大学大学院文学研究科の三根慎二氏より非常に有益な助言をいただきま

した。ここに，心より感謝の意を表します。

注・引用文献

1) Budapest Open Access Initiative.〈http://www.soros.org/openaccess/read.shtml〉［最終確認日：2006-05-12］
2) Lawrence, S. Free online availability substantially increases a paper's impact. Nature. vol. 411, no.6837, 2001, p.521. なお，以下のアドレスから本文が入手可能である。〈http://www.nature.com/nature/debates/e-access/Articles/lawrence.html〉,〈http://www.nature.com/nature/journal/v411/n6837/full/411521a0.html〉［最終確認日：2006-05-12］
3) EPrints. Journal Policies—Summary Statistics So Far.〈http://romeo.eprints.org/stats.php〉［最終確認日：2006-05-12］
4) 独立行政法人科学技術振興機構.〈http://www.jst.go.jp/〉［最終確認日：2006-05-12］
5) 科学技術振興機構. J-STAGE.〈http://www.jstage.jst.go.jp/〉［最終確認日：2006-05-12］
6) 高木元．研究者にとってのセルフアーカイビング . 情報の科学と技術 . vol. 55, no. 10, 2005, p. 434.
7) CiteSeer.IST〈http://citeseer.ist.psu.edu/〉［最終確認日：2006-05-12］
8) Google Scholar Beta.〈http://scholar.google.com/〉［最終確認日：2006-05-12］
9) Lawrence, S.; Giles, C.L.; Bollacker, K. Digital libraries and autonomous citation indexing. IEEE Computer. vol. 32, no. 6, 1999, p.67-71. なお，以下のアドレスから本文が入手可能である。〈http://citeseer.ist.psu.edu/aci-computer/aci-computer99.html〉［最終確認日：2006-05-12］
10) Google Scholar は Beta 版であり，2006 年 5 月の投稿時点での記述である。
11) 三根慎二．“オープンアクセス資料のファイル形式”.〈http://www.openaccessjapan.com/archives/2006/05/oa_1.html〉［最終確認日：2006-05-12］
12) Crane, Diana. 見えざる大学：科学共同体の知識の伝播. 津田良成監訳. 東京，敬文堂，1979, 260 p.
13) Adobe 社. Acrobat family.〈http://www.adobe.co.jp/products/acrobat/〉［最終確認日：2006-05-12］
14) Glyph & Cog. Xpdf.〈http://www.foolabs.com/xpdf/〉［最終確認日：2006-05-12］
15) papy.〈http://homepage3.nifty.com/e-papy/index.html〉［最終確認日：2006-05-12］
16) Ishikawa, O.〈http://ohju.cside4.jp/software/pdftrans/〉［最終確認日：2006-05-12］
17) Taku, Kudo. MeCab: Yet Another Part-of-Speech and Morphological Analyzer.〈http://chasen.org/~taku/software/mecab/〉［最終確認日：2006-05-12］
18) 石田栄美ほか．日本語 PDF ファイルを対象とした学術論文の自動判定．日本図書館情報学会，三田図書館・情報学会合同研究大会発表要綱 2005, 慶應義塾大学，2005-10-22/23，p. 165–168.
19) Department of Computer Science, University of Waikato. Weka.〈http://www.cs.waikato.ac.nz/~ml/weka〉［最終確認日：2006-05-12］
20) Witten, Ian H.; Frank, Eibe. Data Mining: Practical Machine Learning Tools and Techniques. 2nd ed., San Francisco, Morgan Kaufmann, 2005, 525 p.
21) Vapnik, Vladimir N. The Nature of Statistical Learning Theory. 2nd ed. New York, Springer, 2000, xix, 314 p.（SVM に関する日本語文献としては以下の文献が詳しい。Cristianini, Nello; Shawe-Taylor, John. サポートベクターマシン入門．大北剛訳．東京，共立出版，2005, 252p.）
22) Joachims, Thorsten. SVM^{light}.〈http://svmlight.joachims.org/〉［最終確認日：2006-05-12］
23) Chang, Chih-Chun; Lin, Chih-Jen. LIBSVM—A Library for Support Vector Machines.〈http://www.csie.ntu.edu.tw/~cjlin/libsvm/〉［最終確認日 :2006-05-12］
24) Schapire, R.E.; Singer, Y. BoosTexter: A boosting-based system for text categorization. Machine Learning. vol. 39, no. 2/3, 2000, p. 135–168.
25) Dietterich, Thomas G. An experimental comparison of three methods for costructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning. vol. 40, no. 2, 2000, p. 139–157.
26) Freund, Y.; Shapire, R. ブースティング入門．安部直樹訳．人工知能学会誌．vol. 14, no. 5, 1999, p.771–780.
27) Allwein, E.; Schapire, R. E.; Singer, Y. BoosTexter〈http://www.research.att.com/sw/tools/BoosTexter/〉［最終確認日：2006-05-12］
28) Schapire, R. E. The boosting approach to machine learning: An overview. MSRI workshop on nonlinear estimation and classification. 2001, p. 149–172.
29) Graham, Paul “第 8 章「スパムへの対策」”．ハッカーと画家：コンピュータ時代の創造者たち．川合史朗監訳．東京，オーム社，2005, p. 127–135. なお，以下のアドレスから本文が入手可能である。〈http://www.shiro.dreamhost.com/scheme/trans/spam-j.html〉［最終確認日：

2006-05-12］
30) nabeken. bsfilter/bayesian spam filter. 〈http://bsfilter.org/〉［最終確認日：2006-05-12］
31) Robinson, G. A statistical approach to the spam problem. 〈http://www.linuxjournal.com/article/6467〉［最終確認日：2006-05-12］
32) Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. Classification and regression trees. Belmont, Wadsworth International Group, 1984, 358 p.
33) Quinlan, J. R. Induction of decision trees. Machine Learning. vol. 1, no. 1, 1986, p. 81–106.
34) Quinlan, J. R. AI によるデータ解析．古川康一監訳．東京，トッパン，1995, 293 p.

要　　旨

オープンアクセス環境が進展するにつれ，セルフアーカイビングの形式で自らの研究成果を公開する研究者が増加している。そのような成果は，従来のすべてのウェブを対象とする検索エンジンからもアクセスが可能ではあるが，検索結果中の他のものに埋没してしまうことが多い。そこで，本研究ではウェブコンテンツ中からの学術論文，あるいは論文に準ずるコンテンツを判定するシステム構築を目指し，SVM など，多くの手法を用いて自動判定実験を行った。自動判定の手がかりとなる属性群としてはファイル中に出現する語と経験的なルール群を用いた。実験結果からは，段階的な論文判定を行うことで，学術情報専門の検索システム構築が実現可能であることが示唆された。