2004－ＦＩ－74　(11)
2004－ＤＤ－43　(11)
2004／3／26

SVM/HMM による引用文献データの同定

岡田　崇†、高須　淳宏‡、安達　淳‡

論文の引用文献は「著者」、「タイトル」、「雑誌名」といったフィールドから構成される。これらのフィールドに関して、その属性を自動的に同定することができれば有用である。電子図書館は、この情報を利用して引用文献の自動リンク付けを行うことができる。また、このフィールド情報は通常電子図書館に登録し、検索に利用することも考えられる。本研究では、機械学習の手法の１つである SVM を用いてこの自動同定を行う。それに加えて、構文情報を利用して HMM のモデルを作成し、SVM による同定とあわせて適用することで精度向上をはかることができたので報告する。

Identification of citation data by SVM/HMM

Takashi Okada, Atuhiro Takasu, Jun Adachi

Citations of articles are composed of subfields such as ‘author’, ‘title’, ‘journal’ and ‘year’. It is useful to automatically identify attributes of these subfields since they are used for linking a citation with the actual article which the citation indicates. In this article, we employ Support Vector Machine (SVM) which is a method of machine learning to automatically identify subfields. Then, we also apply Hidden Markov Model (HMM) to improve accuracy of identification. Information of subfields identified by SVM and syntactic information analyzed by HMM are integrated for high accuracy identification.

1. はじめに

論文には引用文献が含まれているが、引用文献は関連研究を探したり論文の引用回数を分析したりするのに役に立つ。現在電子化された論文を集めて検索･閲覧を可能にする電子図書館というシステムが大学や研究所などによって構築され、利用することができる。このシステムにおいて引用文献からそれが指し示す論文自体の文書へとリンクがあると、利用者にとって便利である。人手により引用文献の情報を獲得して、それが指し示す論文を電子図書館に蓄積されたデータの中から探し出しリンク付けを行うことは非常に大きな労力を必要とするため、自動的にリンク付けが行えることが望ましい。

引用文献データの同定に関する研究は、2 つの引用文献データが与えられたときに、その文献データが同一の文献であるかどうかを判定する決定問題として定義される。この問題は、任意の引用文献対が与えられたときにその類似度を求める関数と、その類似度に基づいて同一か否かを判定する閾値によって解かれることが多い。ここでは類似度関数の定義と同一性の判定法が重要な課題となる。また、通常は大規模な書誌データベースが作られているため、書誌データベースと引用文献との同定を行う

†　東京大学大学院情報理工学系研究科
‡　国立情報学研究所

ことが多く、大規模データに対して効率的な処理を行うため、インデキシングやブロッキングが実用上重要となる。

類似度関数には、N グラムや編集距離のような近似文字列マッチングの手法が用いられる。近年は、編集距離の重み関数を確率モデルに基づいて決定する方法の研究が進められており、学習用データを用いて編集距離の重み関数を求める試みがなされている[1][2][3]。

引用文献は、通常、著者やタイトルのような書誌項目から構成されている。類似度関数を決定する場合に、これらの書誌項目ごとに類似度の計算法を変えることができれば、よりきめ細かな類似度関数を用いることができ、同定精度の向上が期待できる。図書館情報学の分野で研究されてきた書誌データベース中の重複レコードの発見問題は、引用文献同定と同種の問題で、ここでは、著者名やタイトルを用いた類似度関数が用いられてきた。例えば、Ayres 等は、著者名やタイトルに現れる部分文字列から構成されるコードを用いた重複レコードの発見法を提案している[4]。一方、論文中に現れる引用文献の場合は、書誌項目に区切られていないため、引用文献文字列全体に対して類似度関数を定義するか、もしくは、パーザによって引用文字列を分割した上で類似度関数を定義することになる[5][6]。

引用文献の同一性の判定は、同一である/なしを判定する 2 分類問題ととらえることができる。ここでは、上記の類似度関数を用いて特徴量を求め、統計、機械学習、パターン認識の分野で研究されてきた分類法を用いて、同一性の判定が行われる。引用文献の同定では、SVM を用いた判定法が提案されている[6].なお、引用文献の同定問題は、近年活発に研究されているレコードリンケージの問題と密接な関係がある。この分野の研究動向については、例えば[7]を参照されたい。

我々は、これまでに、大規模書誌データベースに対する引用文献の効率的なマッチング法[8]や書誌項目ベースの類似度関数[3]の研究を行ってきた。本稿では、引用文献の同一性判定を行う分類器の特徴を抽出するための引用フィールドの同定法について述べる。

2. 分類手法

2.1. Support Vector Machine

SVM は Vapnik らによって提案されたパターン認識問題の学習アルゴリズムであり[9]、近似的に汎化誤差の上限を最小化するアルゴリズムであるとみなすことができる。汎化誤差の上限を最小にしようとする原理は Structural Risk Minimization(SRM)原理と呼ばれ、確率の一様収束理論に基づき VC dimension という概念を用いて定式化されている。SRM 原理という理論的背景をもつことは SVM の 1 つの大きな利点であるといえる。SVM は基本的に 2 クラスの分類器であるが、本稿で扱う問題のように、通常の分類問題では多クラスへの分類が必要になることが多い。SVM の多分類問題への拡張手法には 1-vs-rest、1-vs-1、DAG-SVM などいくつかの方法が提案されてきている[10]。

2.2. Hidden Markov Model

隠れマルコフモデルは、確率的な状態遷移と確率的な記号出力を備えたオートマトンである[11]。隠れマルコフモデルは、観測可能な言語データから言語現象の背後にある隠れた構造を推定する場合に有効であり、そのため単語分割モデルや音声認識のための音響モデル、あるいは英語の品詞タグ付け

に使われる統計的言語モデルとしてよく使われる。本研究では、学習データからモデルを作成しておき、テストデータを出力記号系列とみなして、その最適な状態遷移を求めることで、各フィールドの属性を分類する。

3. 提案手法

3.1. 目的

本手法の目的は、1 つの引用文献を提案する切り分け手法によってまず、フィールドごとにわける。そしてそれらの属性を SVM/HMM を併用して用いることで自動的に同定するということになる。

まず本研究で扱う実験データセットについて説明する。本実験では 1 つの雑誌について扱うことにした。この理由として、実際にシステムとして動かす際にはフィールド同定をしようとする引用文献が抽出された雑誌は分かるはずであり、雑誌ごとに別々の HMM モデルを作成したほうが精度は上がるはずだと考えられるからである。つまり 1 つの雑誌について精度がでれば、雑誌ごとにモデルをつくることで多くの雑誌に対応できる。引用文献を取り出した雑誌は、電子情報通信学会の IEICE transactions on fundamentals of electronics, communications and computer sciences で、2000 年度のものをつかっておりそこに掲載されている論文の数は 371 件であり、含まれる引用文献の数は 4651 件である。このうち 4/5 を学習セットに、残りの 1/5 をテストセットとして 5-hold cross validation を行っている。また今回分類の対象とするフィールドは、表 1 のような 9 つとした。表 1 ではフィールド名の下に、そのフィールドに含まれる種類を示した。

表 1　分類対象フィールド

author	Title	journal	Volume	Publisher	day	Month	Year	Other
著者 編者	タイトル 本	雑誌 会議	巻、号、 ページ	出版社	日	月	年	その他

3.2. 切り分け手法

本手法ではじめに行う処理は引用文献をフィールドごとに切り分けるということになる。最も単純に切り分けを行うとすると、単語ごとに分けてしまい、あとから、連続する同じフィールドをつなぐという方法が考えられる。ただし、1 単語のみでは次節で説明する SVM のみの分類が難しくなると考えられるので、本研究ではヒューリスティックな手法でいつくかのデリミタを設定し切り出しを行う手法を提案する。このデリミタで切り分ける手法では、実際のフィールドの分け方により近い切り分けを実現することができ、それが同定精度向上につながると考えられる。

今回の実験では以下のような簡単なデリミタを設定して切り出すことにした。

（1）　基本的には「, 」により切り分けを行う。
（2）　「vol.」「no.」「pp.」「ed.」といったデリミタの前後で切り分けを行う。
（3）　「”」で囲まれている文字列はひとつのフィールドとする。
（4）　前置詞の前で切り分けを行う。

（5）　それぞれの切り分けられた部分の中で先頭や、最後にある数字は切り分ける
（6）　括弧やコロンといった区切り記号で分ける

最終的な評価の際には学習データセットと同じようなフィールドの区切りで分類の正解・不正解を判定する。つまり 2 つに分割されたフィールドはその両方が正しく分類されたときのみ、正解とする。

3.3. SVM による処理

本節では学習データセット及び、前節の手法で切り分けられたテストデータセットを SVM に入力できるようなデータへと変換する手法について説明を行う。SVM に入力するデータはそのデータの特徴を表すような数値ベクトルへと変換しておく必要がある。

処理方針としては、各フィールドに含まれる単語の種類と数をカウントするということになる。つまり、あらかじめ学習データセット全体に含まれるすべての単語の種類を数え上げ、それぞれに番号をふっておき（これを特徴番号とする）ハッシュを作成しておく。そして各フィールドについて、含まれる単語を 1 つずつこのハッシュにあてて、存在すればその特徴番号の値を 1 とするという処理をしていくことで作成することができる。よって出来上がる特徴ベクトルの次元数は学習データセットに存在する単語の数になる。

しかし、この手法では学習データに存在しない単語が、テストデータにあった場合にその情報を使えないという問題点がある。そこで次のような改善を行う。1 つ目は特徴の統合である。これはほぼ同一のものを指すと思われる単語を統合して新たな特徴を作り出すことである。2 つ目は単語の分け方の工夫である。そして 3 つ目は記号文字の除去である。これら 3 つの手法について詳しく説明を行う。

（1）　特徴の統合

例えば、数字を考えると「1」、「5」、「9」などこれらの語が意味するのはほとんどの場合、巻や号、日付などであると考えられる。そしてその場合において「1」と「5」の違いはほとんどないと考えられる。ただし、数字の大きさによって表現するものが異なってくる可能性も考えられるので、同じ桁数の数字を統合するのが妥当だと考えられる。

この例のようにして統合してやれば、例えばテストデータが 1992 のときに学習セット中に 1997 や 1967 はあるが、1992 が存在しなかったとしても、1992 が 4 桁の数字と変換されれば、学習セット中にも 4 桁の数字があるのでその特徴をもつとみなすことができるようになる。

（2）　単語の分け方の改善

単語を分ける際、スペースがその区切りになるとは限らない。つまり記号などが単語の区切りになることもあるのでそれを考慮する。また、記号を単純に区切りと見なしてしまうことには問題がある場合もある。例えば、ハイフンで単語がつながっている場合、それは複合語である可能性もあるし、あるいは、単純に区切り文字である場合もある。よって下に示した例のように元々の形を残しつつ、それ以外にさらに記号をスペースに変換したものも付加するという処理でこの問題を解決する。この手法はスラッシュやコロン、略称番号などに適用する。

Ex.1) multi-task → multi-task multi task

Ex.2) SVM-support vector・・・→ SVM-support SVM support vector・・・

（3）　記号文字の削除

記号を取り除けば他の単語と一致するものも多いのでこういった記号は削除することにする。たとえば、単語を強調するために使われるシングルクォーテーションなどがこれに当たる。

このデータをもとに SVM での分類を行う。本研究では多分類を行う必要があり、前述したようにSVM の多分類の拡張手法には様々な手法が提案されてきているが、ここでは、事前に同様のデータを用いて行った予備実験で最もよい精度を示した 1-vs-rest を用いることとした。また、この予備実験にてカーネルによる差はほとんどでなかったため、パラメータの数が少ない linear kernel を用いる。

3.4. HMM による処理

SVM のみを用いた手法は個々のフィールドに関する特徴のみを用いている。よって例えば、「1998」というものがあったとき、これがページなのか年号なのか判定するのは難しい。それ以外にも、「Neural Networks」などはタイトルと雑誌名どちらにもでてくるので、このようないくつかのフィールドで同じ物が出てくる場合 SVM では判定しきれない。そこで構文情報を利用することが考えられる。引用文献の書き方には必ずしも 1 通りの書き方で書かれているわけではないが、著者の次はタイトルが来ることが多い、あるいは年号は 1 番最後に書かれる場合が多いなど、引用文献中のフィールドの語順に関する情報を用いることも有用であると思われる。そこでこのような構文情報を各フィールドの同定に利用するため、隠れマルコフモデルを SVM と併用して用いる。

ここで、具体的な適用手法について説明する。まずは SVM を用いて各フィールドの同定を行う。そしてフィールドごとに、それぞれのクラスの SVM 分類器の出力を成分にもつ

$$\vec{v} = (author, title, journal, volume, day, month, year, publisher, other)$$

のようなベクトルを作成する。そして、1 つの引用文献には、そのようなベクトルを各フィールド分並べた、$\vec{v}_1\vec{v}_2\vec{v}_3\vec{v}_4\vec{v}_5\vec{v}_6$ のようなもの（1 つの引用文献が 6 つのフィールドから成る場合）が割り当てられることになり、個々の引用文献ごとに、このベクトル列が隠れマルコフモデルから出力されたとしてその状態遷移を推定する。

状態としては「著者」、「タイトル」といった各フィールドの属性とする。そして出力記号は $\vec{v}_1$ や $\vec{v}_2$ といった各フィールドに割り当てられたベクトル（次元数は分類するフィールドの数）である。状態遷移確率は、「著者」という状態から「タイトル」という状態へのものであれば、著者の次にタイトルが存在する頻度をカウントすることで決定する。すなわち、遷移確率により引用文献中の各フィールドの位置情報が反映される。記号の出力確率は、まず状態ごとに n 次元のガウス分布を出力として仮定する（n = 分類するフィールド数）。そして状態ごとに属するべきフィールドの訓練データを利用して平均と分散を計算することで分布が得られる。この分布にテストデータを当てることで尤度を算出する。ただし、ガウス分布を尤度の分布と見なすと、距離が平均以上になると逆に尤度が下がるようになってしまう。そこで実際の実験ではこれに関する補正を施している。

このような方法で SVM と隠れマルコフモデルを併せて利用することでフィールドの特徴と位置情報を属性の同定に使うことができるので、精度向上をはかれるのではないかと考えられる。

3.5. OCR データへの適用手法

最後に OCR データへの適用手法について説明を行う。OCR の認識では、「１」と「l」の間違いや

「0」と「O」の間違いなど良く似た文字を誤認してしまうことがある。電子図書館に蓄積する電子化された論文には、元々電子データの論文だけでなく紙に印刷されたものしか手に入らず OCR で認識しなければならないものもあるので、OCR データに本手法が適用できるかということも重要である。そこでこのような誤りを含む OCR データへ本手法を利用する際に行うべき改善法について説明をする。

本手法を OCR データに適用する際に問題になるのは、特徴ベクトル化の部分だけである。特徴ベクトル化における問題点は、テストデータに含まれる単語が学習データセット中にない場合、その単語は全く考慮されないということである。そこで編集距離を利用することを考える。編集距離とは文字列間の距離を表すもので、一方から他方を得るために削除、挿入、変更という操作を何回行う必要があるかにより求められる。本手法の中では、各テストデータを特徴ベクトル化する際、テストデータ中に含まれる単語それぞれに対して、この編集距離をすべての学習セット中の単語に対して計算しある閾値以内の編集距離にある単語とマッチしたとみなすように変更することで、前述した問題を改善できると考えられる。

4. 実験結果

SVM のソフトとしては SVM*light*[12]を利用して、多分類に拡張して実験を行っている。まず予備実験として、特徴ベクトル化に上述した改善を加える場合と加えない場合の比較を図 1 に示した。これは recall について示している。以降すべての結果は recall についてのみ示している。この結果を見ると確かにこの改善手法が有効であることがわかる。また、各テストデータに割り当てられる特徴ベクトルの長さを、この改善を加える場合と加えない場合について比較すると、表 2 のように改善を加えるほうが確かに長くなっており、分類を行う際の情報量の増加が精度向上につながったといえる。

表 2　特徴ベクトルの長さ

	set1	set2	set3	set4	set5
改善を加えない	1.895662	1.887812	1.889399	1.915559	1.929984
改善を加える	2.222096	2.22515	2.224012	2.241575	2.258286

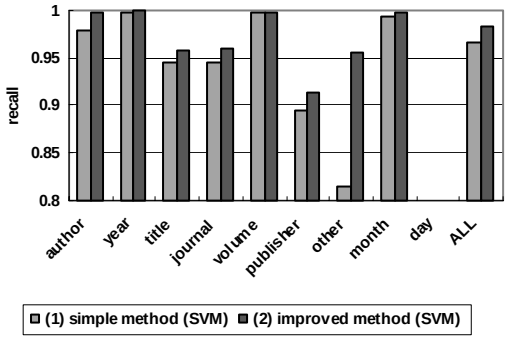

図 1　特徴ベクトル化の比較

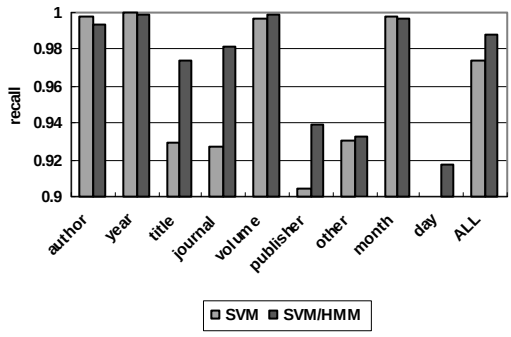

図 2　SVM のみと SVM/HMM の比較

次に本研究の主目的である 1 つの引用文献からフィールドを切り分け、その属性を同定する実験結果について説明する。最初に SVM のみによる同定実験と、SVM/HMM による同定実験の結果を比較

する。図 2 を見ると HMM を使う場合では、title、journal、publisher、day で精度向上が見られる。また、全体的にみても 0.974 から HMM を使うことで 0.988 に上がっており、HMM を使うことで確かに精度を高めることができた。

さらに、本稿で提案する切り分け手法の優位性を確かめるために、デリミタで切り分けを行う手法と、単純に単語で分けた手法の比較も行った。図 3 にこれを示す。全体で比較すると単語分けでは、0.874 にまで下がってしまい、デリミタ分けの優位性は明らかである。また個々のフィールドについて見てみると、year、volume、month、day という元々1 単語からなるフィールドでほぼ同程度の精度であるが、それ以外のフィールドではデリミタ分け手法が大きな優位性を示した。これは、やはり細かく切ることで、情報が少なくなりすぎることが判定の間違いにつながっていると考えられる。

最後に認識誤りを含む OCR データに本手法を適用した実験について説明する。まず今回扱った OCR データであるが、上で扱ったものと同じ論文誌のものの一部を読み取ったものである。合計件数は 4339 件である。また、OCR の認識精度は 0.992 である。

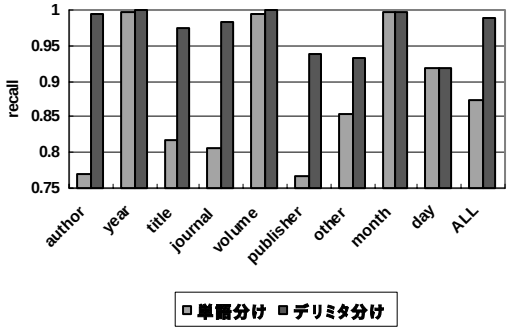

図 3　単語分けとデリミタ分けの比較

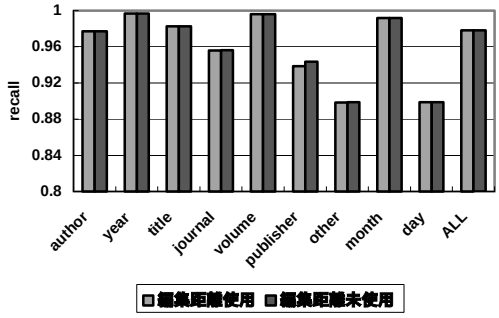

図 4　OCR データ編集距離の使用比較

図４では編集距離を使う場合と使わない場合の 2 種類の方法を比較している。大きな違いはでなかったものの、編集距離を使う場合の方が良い結果を示した。向上の割合が小さかった原因として OCR 認識精度がかなり高かったことが考えられる。1 つのフィールド中で複数の単語が誤りを含んでいることが少なく、一致する単語だけである程度精度が出たのではないかと思われる。全体の精度をみると 0.978 と高い精度をだすことができた。

5. まとめ

本研究では、引用文献中の著者、タイトルといったフィールドを同定する手法を提案した。さらに紙に印刷された形でしか手に入らない論文にも対応できるよう、文字の認識誤りを含むような OCR データに対しても適用できるような改善手法についても説明した。結果として電子データの切り分け･分類実験で 0.988、OCR データの切り分け･分類実験で 0.978 と高い分類精度を示すことができ、書誌情報を自動的に作成するために実用できると考えている。

今後の課題は提案システムを実用化させることである。実用化するためにはいくつか解決すべき問題がある。まず 1 つ目として日本語引用文献への適用が挙げられる。これには、引用文献を単語へと切り分ける処理で形態素解析を行う必要がある。その後の処理は基本的に英語と全く同様に扱えると

考えられるので、手法の適用は比較的に簡単にできると考えられる。2 つ目として、分野による引用文献の書き方の違いに対応する必要もあると考えられる。今回の実験では電子情報通信学会の論文誌に掲載された論文の引用文献のみについて実験を行っている。そのため、URL やソフトウェアなどを表記したものは論文を表すものと大分書き方が違うものの、論文に関する分は概ね同じような書かれ方をしている。しかし、分野が違うと全く書き方が変わることもある。例えば日本物理学会誌では、多くの場合「タイトル」は省略される。よって HMM のモデルが論文誌により大きく違うことが考えられる。このように大きく順番が異なるものに対して対応するため、分野ごとに別々に HMM のモデルを作成しておき、引用文献データの同定を行う前にその論文の掲載誌でまず分類して、それに従って別々のモデルへ入力し判定することが有効であると考えられる。また、システムとして考えたときに付加するべき機能として、本手法によりフィールドが同定されたデータをすでに登録されたデータベースにあてることで、OCR 認識誤りがあった場合にはそれを訂正することができればよいだろう。

参考文献

[1]E. S. Ristad, P. N. Yianilos:Learning string-edit distance,IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 20, No.5,pp.522-532, 1998.

[2]M. Bilenko, R. J. Mooney:Adaptive Duplicate Detection Using Learnable String SimilarityMeasures,Proc. 9th ACM Intl. Conf. on Knowledge Discovery and Data Mining,2003.

[3]高須淳宏、相原健郎：「テキスト認識エラーモデルによる引用文献文字列からの書誌要素の抽出」電子情報通信学会論文誌, Vol. J87-D-II, No. 6 掲載予定 (2004)

[4]F. H. Ayres, J. A. W. Huggil, E. J. Yannakoudakis:The Universal Standard Bibliographic Code (USBC): Its use for clearing, merging and controlling large databases,Program - Automated Library and Information Systems, Vol.22, No.2,pp.117-132,1988.

[5] S. Lawrence, C. L. Giles, and K. Bollacker, “Digital Libraries and Autonomous Citation Indexing,” IEEE Computer, 32(6), 67-71, 1999.

[6] 伊藤 敬彦, 堀部 史朗, 新保 仁, 松本 裕治, “ 複数尺度を用いた参考文献の同定 ”, 情報処理学会研究報告, 2003-DBS-130, pp181-188, 2003.

[7]相澤彰子、高須淳宏、大山敬三、安達淳：「異種データベース間でのレコード照合に関する研究動向」NII Journal, No. 8, 掲載予定 (2004).

[8]高須淳宏、片山紀生、大山敬三、安達淳、影浦峡：「学術文献画像の書誌情報の近似マッチング法」情報処理学会論文誌：データベース, Vol.42, No.SIG 1, pp.148-158, 2001.

[9] B. Scholkopf, “Support Vector Learning,” PhD thesis, Universitat Berlin, Germany, 1997.

[10] C. Hsu and C. Lin, “A comparison on methods for multi-class support vector machines. Technical report,” National Taiwan University, Taiwan, 2001.

[11] 北 研二, 言語と計算－4 確率的言語モデル, 東京大学出版会.

[12] http://svmlight.joachims.org/