音楽記事中のアーティスト名を対象としたエンティティリンキング

浦田 智昭　　前田 亮

立命館大学情報理工学部 〒525-8577 滋賀県草津市野路東 1-1-1

E-mail: is0156ss@ed.ritsumei.ac.jp, amaeda@is.ritsumei.ac.jp

あらまし

音楽情報を取得するために，Web 上で音楽記事やブログを読む機会が増えてきている．しかし，文書中に出現したアーティストに読者が興味を持ち，詳しい情報を知りたいと思っても，アーティストに関する情報へのリンクが存在しない場合が多く，Google などの検索エンジンで逐一調べるのは面倒である．そこで本研究では，読者が文書中に出現するアーティストの情報を検索する負担を軽減させるために，文書中に出現するアーティスト名などのエンティティを抽出し，そのアーティスト情報を得ることができる Web サイトと自動リンクするエンティティリンキングの手法を提案する．本手法は 2 つの手順に分けられる．1 つ目は，アーティスト名の抽出である．アーティスト名は固有名詞であり，ほかの固有名詞と差別化しアーティスト名の抽出を行う必要があるので，あらかじめ音楽記事から学習データを用意し，Support Vector Machine による機械学習を用いてアーティスト名の抽出を行う．2 つ目は，アーティストに関連する Web サイトの選択である．また，アーティストの関連リンクは，Wikipedia 記事へのリンクを使用する．提案手法の有効性を検証するため，日本語の音楽記事 35 件に対して手動でアーティスト名を抽出したものを正解データとして，交差検定による評価実験を行った．再現率が 0.2788，適合率が 0.7530，F 値が 0.4070 となった．また，音楽記事 10 件に対し Wikipedia の記事タイトルとの編集距離を求め正解記事をリンクする実験を行い，正解率は 0.8740 となった．

キーワード　固有表現抽出，機械学習

1. はじめに

昨今，音楽情報を取得するために音楽記事や音楽ブログを読む機会が増えてきている．しかし，文書中に読者が興味のあるアーティスト名が出現してもアーティスト名のリンクが存在しないことが多く，逐一 Google などの検索エンジンを利用することによってアーティストの情報を調べなければならず，非常に面倒である．そこで，本研究では読者が音楽に関する情報を検索する負担を軽減させるために，文書中に出現するアーティスト名などの固有情報を Support Vector Machine（SVM）による機械学習を用いて抽出し，アーティスト名を Wikipedia の記事等に自動リンクすることによって，読者がアーティスト情報を収集する際により利便性を向上させ，負担を軽減する手法を提案する．提案手法のイメージを図 1 に示す．音楽記事中では「ももいろクローバーZ」と「ももクロ」がアーティスト名として出現するが，このアーティスト名に Wikipedia の「ももいろクローバーZ」の Web ページのリンクを付与する．しかしこのイメージ図は最終目標であり本提案手法ではこのような略称の例を抽出することはできない．

エンティティリンキングとは，テキスト中で何らかの「実体（エンティティ）」を指示する記述を抽出して，外部のデータベースや知識ベースの項目と対応づける操作のことである．エンティティリンキングに関する研究は，情報検索や自然言語処理の分野で活発に行われており，研究対象は主に新聞記事や Web 文書が対象とされている．これらのテキストでは，人名や組織名などの固有表現がエンティティとして重要な役割を持つことから，エンティティリンキングでは固有表現が重視される場合が多い．従来の研究は対象テキストからエンティティを抽出する際，辞書を用いてエンティティを抽出する手法と機械学習を用いてエンティティの抽出を行う手法があるが，本研究では機械学習を用いた手法を用いてエンティティを抽出する．

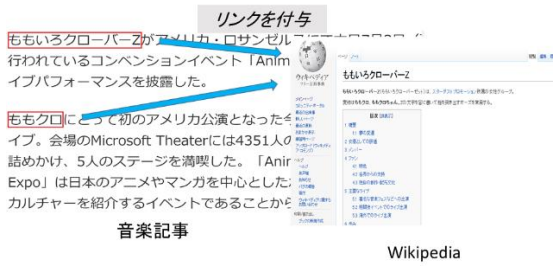

図 1 ：提案手法のイメージ図

固有表現抽出を行う研究において，人手で作成した規則に基づく手法や，統計的機械学習を用いた手法が提案されているが，人手で作成したルールに基づく手法[2]よりも，機械学習を用いた手法[3]のほうが精度が良いことが示されている．本研究では，SVM を用いて単語単位でアーティスト名の抽出規則を自動的に学習する手法を用いてアーティスト名の抽出を行う．Wikipedia はオープンコンテント百科事典であり，だれでも自由に無料で編集することができ，世界各国の様々な言語で公開されている．2016 年 1 月時点で 290 以上の言語で執筆されており，総記事数は 3,400 万以上の大規模な百科事典である．Wikipedia の記事数は膨大であるだけでなく，記事の質も高いとされているため，読者が興味のあるアーティストの情報を調べるのに有用な手段の一つとして考えられる．

2. 関連研究

2.1. エンティティリンキング

エンティティリンキングとは，テキスト中で何らかの「実体（エンティティ）」を指示する記述を抽出して，外部のデータベースや知識ベースの項目と対応づける操作のことであり，Wikipedia が知識ベースの一つとしてよく用いられる．エンティティの対象物は主に，人名，組織名，団体名，地名などがあるが，本研究では，アーティスト名をエンティティとして取得する．

古川[1]らは日本語の技術的な文書中の専門的な用語から，英語 Wikipedia 項目へリンクし，そのために必要な語義曖昧性解消手法を提案している．用語抽出には，与えられた日本語の学術文献の記事から辞書を用いて専門用語を抽出している．知識ベースは英語版 Wikipedia 記事の記事およびセクションから主題，副題，URL，文書，内部リンクを抽出して構築している．用語抽出の際に辞書を用いると，用語に表記ゆれや辞書に存在しない略称が存在した場合，抽出が正しくされない場合がある．そこで，本研究ではアーティスト名抽出を行う際に辞書を用いるのではなく，機械学習による手法を用いることでアーティスト名抽出を行うことによって，用語抽出の際に表記ゆれがあっても抽出する手法を提案する．

2.2. 固有表現抽出

固有表現抽出は，地名・人名・組織名などの固有名詞や日時・時間・通貨などの数値表現を抽出する技術である．IREX（Information Retrieval and Extraction）というワークショップの日本語固有表現タスクでは，組織名，人名，地名，固有物名，日付表現，時間表現，金額表現，割合表現の 8 種類の固有表現を定義している．固有表現を抽出する手法は，主に二つに分かれる．一つ目は人手で作成した法則に基づく手法[2]であり，二つ目は機械学習による手法である．機械学習の手法として，SVM（Support Vector Machine），最大エントロピー法，条件付き確率場に基づく手法などが提案されている．機械学習を用いた固有表現抽出では，入力文を解析単位（トークン）に適当に分割し，固有表現を構成する一つもしくは複数のトークンをまとめる手法が一般的である．

山田ら[3]は SVM を用いた固有表現抽出を行っている．単語を解析の単位として，単語自身，品詞分類，文字種などを素性として利用している．実験には CRL（郵政省通信総合研究所）固有表現データを使用している．CRL 固有表現データは，毎日新聞 95 年度版 1,174 記事，約 11,000 文に対して固有表現が付与されている．提案手法による抽出実験では，F 値で約 83%という高い精度の結果を得ている．このことから，固有表現抽出において SVM が有用であることが示されている．

この研究を含めて，前後 2 トークンの情報を素性として使用することが一般的である．しかし，固有表現の構成要素数が多い場合は十分な素性が与えられず，解析誤りが起こりやすくなる．ここで，中野ら[4]は，文節区切りを行い文節内外の情報を素性として使用する手法を提案している．CRL 固有表現データを用いた固有表現抽出の実験結果は，F 値で 89%という結果を得ている．

本研究では山田らによる SVM を用いた抽出手法，中野らの文節内外の素性を利用した手法を参考に，SVM を用いて単語単位で解析することでアーティスト名の抽出を行う．アーティスト名は人名に属する固有表現であるので，与える素性の記号を従来の手法より細分化して与えて学習させる．

3. 提案手法

本研究の提案手法では，音楽記事群からアーティスト名を抽出し，そこからアーティスト名の学習モデルを作成し，アーティスト名の抽出を行い，アーティストの関連リンクを取得する．

本手法は二つの手順に分かれる．一つ目は，アーティスト名候補の抽出である．アーティスト名は固有名詞であり，本手法で抽出したい用語であるので，ほかの固有名詞と差別化し固有表現抽出を行う必要がある．よって，あらかじめ音楽記事から学習データを用意し，Support Vector Machine(SVM)による機械学習を用いてアーティスト名の抽出を行う．二

つ目は，アーティストに関連する Web ページへのリンクの取得である．そこで，アーティストの関連リンクとして，Wikipedia のアーティストの記事を用いることによって情報を得る．本研究の提案手法の処理の流れを図 2 に示す．図 2 の説明は 3.1 節より行う．

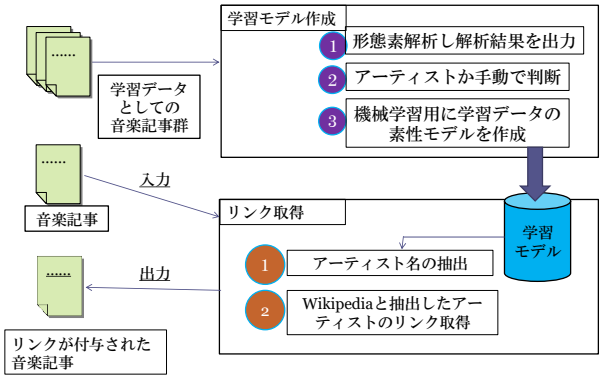

図 2: 提案手法の処理の流れ

3.1. 学習モデル作成

学習モデル作成は三つの手順を踏む．

一つ目は学習モデル作成のために取得してきた音楽記事群のデータを形態素解析することである．形態素解析器を通して得た音楽記事群の形態素解析結果を取得する．

二つ目は音楽記事の形態素解析結果から，アーティスト名を手動でタグ付けを行う．表 1 は形態素解析された文書中に出現するアーティスト名を手動で判定した結果の一部である．表は形態素ごとに区切られていて，「タグ」は 1 か 0 かでアーティスト名かそうでないかのタグであり，1 はアーティスト名の一部，0 はアーティスト名以外という方法でタグ付けを行っている．「形態素」は文書中に出現した単語，「品詞」は形態素解析によって得た品詞，「未知語」は形態素解析器の辞書にその単語が含まれているかどうかを形態素解析器が出力した結果である．1 が未知語で 0 が辞書に登録されている単語であることを表している．

三つ目は機械学習に用いる素性を与える．素性とは，識別に使う情報のことをいう．

本研究で与える素性は，アーティスト名のタグ，品詞細分類，文字種，未知語判定情報の 4 種類である．品詞には，品詞大分類と品詞細分類の 2 種類存在し，品詞大分類とは，名詞，動詞，助詞などの分類であり，品詞細分類とは，「名詞，固有名詞，地域，国」，「記号，読点」などの品詞の細かい分類である．素性には，より多くの情報を含む品詞細分類を用いる．文字種は“カタカナ”，“ひらがな”，“漢字”，“記号”，“数字”，および“アルファベット”の 6 種類のいずれかを指す．

表 1：アーティスト名のタグ付け

タグ	形態素	品詞	未知語
0	初日	名詞,一般,*,*	0
0	が	助詞,格助詞,一般,*	0
0	米国	名詞,固有名詞,地域,国	0
0	の	助詞,連体化,*,*	0
1	フー	名詞,固有名詞,一般,*	0
1	・	記号,一般,*,*	0
1	ファイターズ	名詞,固有名詞,組織,*	1
0	、	記号,読点,*,*	0
0	2	名詞,数,*,*	0
0	日	名詞,接尾,助数詞,*	0
0	目	名詞,接尾,一般,*	0
0	が	助詞,格助詞,一般,*	0
0	英国	名詞,固有名詞,地域,国	0
0	の	助詞,連体化,*,*	0
1	ミューズ	名詞,一般,*,*	0
0	、	記号,読点,*,*	0

表 1 の例では「ファイターズ」という単語を学習するとき，品詞細分類は「名詞，固有名詞，組織，＊」で，このとき素性として「名詞，固有名詞，組織，＊」という文字列を ID に変換して素性としている．文字種はカタカナ，未知語判定情報では未知語であるという素性を学習機に与えることになる．また，アーティスト名は複数の形態素によって構成されている場合が多く，素性の単語自身を学習させるだけでは不十分である．そこで，前後の形態素の情報を与えることが必要である．一般的には前後 2 形態素を与えるが，最適な前後の形態素数の検証のため，素性に用いる前後の形態素数を変化させることによって精度の比較を行う．以上の三つの手順を踏み，学習モデルを作成する．

3.2. アーティスト名の抽出

タグがつけられていない音楽記事を入力し，3.1 節で作成した学習モデルを使用し，SVM によって音楽記事中のアーティスト名を含む形態素を推定した結果を出力する．出力結果の例を表 2 に示す．表 2 の各項目は左端のカラムを除き表 1 と同様である．推測結果は表 2 の推定結果の項目にあり，入力したテストデータの音楽記事中の形態素をアーティストの一部と推測すればタグは 1，アーティストの一部

ではない形態素と推測した場合，タグは 0 という結果が出力される．

表 2： アーティスト名を含む形態素の推測結果

推測結果	形態素	品詞	未知語
0	初日	名詞,一般,*,*	0
0	が	助詞,格助詞,一般,*	0
0	米国	名詞,固有名詞,地域,国	0
0	の	助詞,連体化,*,*	0
1	フー	名詞,固有名詞,一般,*	0
1	・	記号,一般,*,*	0
1	ファイターズ	名詞,固有名詞,組織,*	1
0	、	記号,読点,*,*	0

3.3. リンク取得

学習モデルをもとに表 2 のようにアーティスト名として抽出された形態素のタグの推測結果が 1 が連続して一つのアーティスト名として推測している場合が存在する．そのとき形態素を連結し一つのアーティスト名として文字列を作成する．例えば表 2 の推測結果では「フー」と「・」と「ファイターズ」がアーティスト名の形態素として推測されているので，これらの形態素を連結して「フー・ファイターズ」とアーティスト名の一つの文字列としてまとめる．形態素を連結して得たアーティスト名と Wikipedia の全記事タイトルをマッチングすることによってアーティスト名のリンクを取得する．また，抽出されたアーティスト名は表記ゆれの可能性が考えられるため，抽出されたアーティスト名と Wikipedia のタイトルの編集距離を計算し，リンクを取得する．編集距離とは，一文字の挿入・削除・置換によって，一方の文字列をもう一方の文字列に変形するのに必要な手順の最小回数のことである．例えば，記事中に「ボン・ジョビ」というアーティストが出現したとき，そのアーティストに対応する Wikipedia のタイトルは「ボン・ジョヴィ」である．「ボン・ジョビ」を「ボン・ジョヴィ」にするためには「ビ」を「ヴ」に置換し，「ィ」を挿入する二つの編集を行う．よって編集距離を用いれば 2 でアーティスト名の正しい Wikipedia の記事が取得できる．

4. 評価実験・考察

本章では，提案手法の有効性を検証するための評価実験とその考察について述べる．評価実験には，YAMAHA のウェブサイト[1]から収集した音楽記事 35 記事を用いて，交差検定することによって提案手法の有効性を検証した．音楽記事は邦楽・洋楽の記事を対象にしており，記事の文字数は最長で 8,966 文字，最短は 1,932 文字，平均文字数は 4,313 文字，1 記事中のアーティストの出現数の平均は 34 アーティストである．

4.1. 使用ツール

4.1.1 Kuromoji

本実験では形態素解析器は Kuromoji[2]を使用した．Kuromoji は Java で記述されているオープンソースの日本語形態素解析エンジンである．Kuromoji は基本的には MeCab-IPADIC 辞書をサポートしており，形態素解析結果は MeCab と同様の結果が出力される．

4.1.2 LIBLINEAR

本実験では，SVM の利用に LIBLINEAR[5]という台湾国立大学の Chih-Jen Lin 教授のチームが公開しているオープンソースの機械学習パッケージを用いる．LIBLINEAR は線形予測に特化しており，同チームが公開している LIBSVM と比べはるかに高速に動作するため，大規模なデータへの利用に向いている．

4.2. 評価方法

提案手法による抽出結果と手動で抽出したアーティスト名を比較することによって，正解を判定する．その正解数と実験による抽出数から適合率，手動で抽出したアーティスト数から再現率を算出し，それらから F 値を算出した．F 値とは，適合率と再現率の調和平均のことである．適合率 *precision*，再現率 *recall*，F 値 *F-measure* は次の式で算出する．

$$precision = \frac{R}{N}$$

$$recall = \frac{R}{C}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}$$

[1] http://www.yamaha.co.jp/ongakukiji/
[2] https://www.atilika.com/en/

R は正解したアーティスト名の抽出結果の形態素数，N は実験による抽出結果の形態素数，C は手動で抽出した正解のアーティスト形態素数を表している．ここでアーティスト数ではなく，アーティスト名の形態素数を単位としているのは，アーティスト名の抽出結果においてアーティスト名の一部が抽出された結果が多く，評価にかかる作業コストを軽減するためである．

また，実験に際して交差検定の手法の一つである leave-one-out 交差検定を行った．leave-one-out 交差検定とはデータを k 個用意し，一つをテスト事例とし，残る k-1 個を訓練事例として k 回検証を行い，k 回の結果を平均して一つの推定結果を得る手法である．今回 k を 35 記事，訓練 34 記事，テスト 1 記事の比率で実験を行った．

4.3. 素性の比較実験

4.3.1 素性に適切な形態素数

固有表現を機械学習する際，一般的には前後 2 形態素までの情報を素性とすることが多いが，ここでは適切な前後の形態素数を検証するための実験を行った．使用する素性の前後の形態素数を一つから四つの範囲に変化させた際の再現率，適合率，F 値を表 3 に示し，またそれぞれの値の最大値には太字で示す．

表 3：素性の前後の形態素数による精度の違い

前の形態素数	後ろの形態素数	再現率	適合率	F 値
1	1	0.1963	0.4819	0.2790
1	2	0.2546	0.4957	0.3364
1	3	0.2432	0.5336	0.3341
1	4	**0.2686**	**0.6529**	**0.3807**
2	1	0.1793	0.4442	0.2555
2	2	0.2764	0.5898	0.3764
2	3	0.2096	0.5574	0.3046
2	4	0.2210	0.5557	0.3163
3	1	0.2118	0.4824	0.1326
3	2	0.1996	0.5476	0.2926
3	3	0.2196	0.5914	0.3202
3	4	0.2487	0.5856	0.3491
4	1	0.1800	0.5164	0.2670
4	2	0.2332	0.5830	0.3331
4	3	0.2380	0.5723	0.3362
4	4	0.2181	0.5300	0.3090

4.3.2 素性の比較実験

海外のアーティスト名が音楽記事中に出現するとき，グループ名の単語の区切りや，個人名の姓と名の区切りなどに中点がよく使われる．そこで，アーティスト名を抽出するとき中点の情報を素性に加える．4.3.1 節の実験では，中点の情報を認識する際に学習機は単なる「記号」という品詞種別としてしか扱っていなかったが，中点そのものの情報を品詞種別「中点」として品詞 ID に追加し学習することで実験の結果にどのように影響するか実験した．使用する素性の前後の形態素は表 3 の F 値が最も高かった前後の形態素数上位 3 件を使用した．この実験の結果を表 4 に示す．また，中点の素性を与える前と与えた後の変化を見るため，前 3 つ，後ろ 4 つの形態素を使用した結果を図 3 に示し，最も精度が高かった値を太字で示す．

表 4：中点の情報を与えた実験結果

前の形態素数	後ろの形態素数	再現率	適合率	F 値
1	4	0.2784	0.6288	0.3860
2	2	0.2236	0.6551	0.3334
3	4	**0.2788**	**0.7530**	**0.4070**

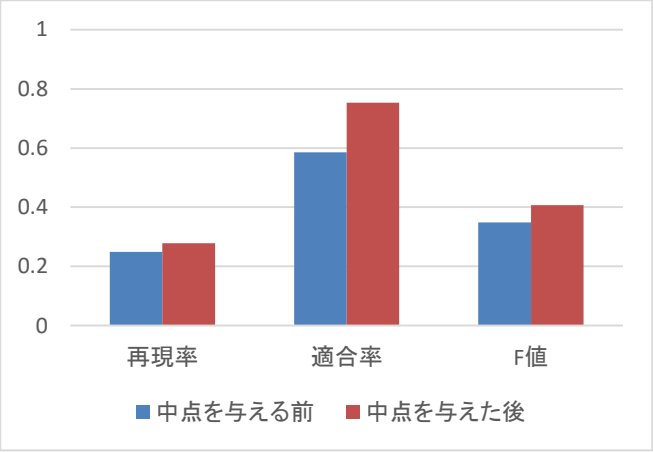

図 3：中点の素性を与える前後の結果
(前 3 つ，後ろ 4 つ)

4.3.3 考察

使用する前後の素性の情報は一般的には前後 2 つの形態素の情報を使用するが，表 3 より前 1 つ，後ろ 4 つの形態素の情報を与えてアーティストを抽出した結果が F 値が最も高かった．後ろの形態素を多く与えると F 値が向上し，逆に前の情報を多く与え

ていくにつれてF値が低下する傾向が見られた．全体的に再現率が低いが，これは再現率の計算時に手動で抽出したアーティストと正解したアーティストの形態素数が完全一致した場合をカウントしたため，再現率が低くなったと思われる．

また，完全一致したアーティスト名の形態素数を増やすために，中点の素性を与えて実験をした．表4，より前3つ，後ろ4つの形態素情報を使用した結果が再現率が0.2788，適合率が0.7530，F値が0.4070と表3のどの結果よりF値が高い．中点の素性を与える前より，中点の素性を与えた後の数値のほうが高いことから，中点の素性情報は精度向上に効果があると考えられる．

部分一致の場合に，再現率を上げる効果があることを検証するため，表4の前後形態素数に部分一致で評価を行った場合を表5に示す．また，完全一致，部分一致の再現率，適合率，F値の前の形態素数3つ，後ろの形態素数4つのときの変化のグラフを図4に示す．図4より部分一致の場合，再現率が高くなったので，完全一致のときに取りこぼしていた形態素が取れるようになるといえる．

表5：正解した形態素が部分一致のとき

前の形態素数	後ろの形態素数	再現率	適合率	F値
1	4	0.6572	0.6288	0.6426
2	2	0.5439	0.6551	0.5943
3	4	0.6557	0.7530	0.7010

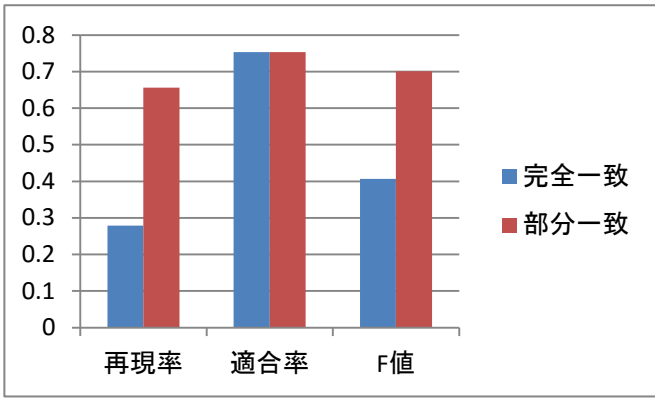

図4：完全一致・部分一致のときの評価結果の違い(前3つ，後ろ4つ)

4.4 エンティティリンキング実験

4.4.1 編集距離を用いた実験

4.1節で使用した音楽記事データの中から10記事を用いて，記事中のアーティスト名を抽出しWikipediaの全記事タイトル2,863,283個と編集距離を用いてマッチングした．10記事中のアーティスト数は357組で，編集距離の範囲を0から2まで変化させていき正解数がどの程度変化するのか実験を行った．なお，編集距離0はアーティスト名とWikipedia記事タイトルが完全一致であることを表している．正解は記事中のアーティスト名がWikipediaのタイトルに存在した場合，記事がそのアーティストのものかどうかにかかわらず正解とみなす．マッチングの実験結果の編集距離による正解率の変化のグラフを図5に示す．

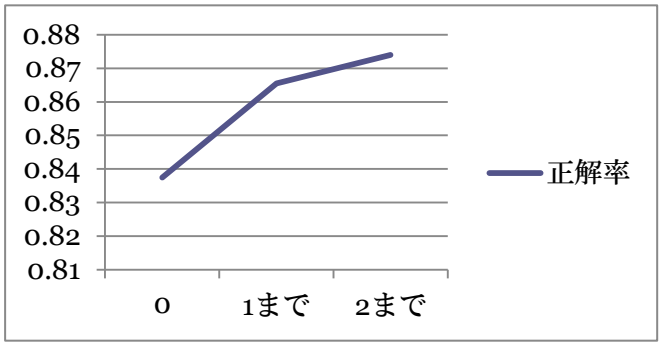

図5：編集距離による正解率の変化のグラフ

4.4.2 考察

図4より編集距離が0のとき正解率が0.8375となり，編集距離の範囲を1までとしマッチングを行った結果，正解率が0.8655となり，範囲を2までとした場合，正解率が0.8740となった．正解率は編集距離が0から1までとしたとき，0.028増加し，1から2までとしたとき，0.0085増加した．編集距離の範囲を1までとしたときにマッチングした結果の例を表6に示す．

表6：正解抽出の例（編集距離1まで）

アーティスト名	Wikipediaタイトル名
スティーヴィ・レイ・ヴォーン	スティーヴィー・レイ・ヴォーン
エース・オブ・ベイス	エイス・オブ・ベイス
KoЯn	Korn

表5より編集距離が1までの場合は「スティーヴィ・レイ・ヴォーン」，「エース・オブ・ベイス」のように音楽記事中のアーティスト名に長音「ー」の表記ゆれが存在したときに正解が抽出され，また「KoЯn」のようにアーティスト名に特殊文字が存在したときに正解が抽出された．次に編集距離の範

囲を 2 までとしたときにマッチングした結果，正解が抽出できた例を表 7 に示す．

表 7：正解抽出の例（編集距離 2 まで）

アーティスト名	Wikipedia タイトル名
ONE OK ROCK	ONE_OK_ROCK
ノヴェンバー	ノーベンバー
スノーウィ・ホワイト	スノウィー・ホワイト

表 7 より編集距離が 2 までの場合は「ONE OK ROCK」のように音楽記事中のアーティスト名にアルファベット表記で空白が単語間に存在している場合，正解が抽出された．これは，Wikipedia のタイトル内ではアルファベット表記のアーティスト名が出現した場合，単語間の空白の位置にアンダーライン「_」が代わりに挿入されているのが原因だと考えられる．また，「ノヴェンバー」や「スノーウィ・ホワイト」のように音楽記事中のアーティスト名に長音「ー」，「ベ」の代わりに「ヴェ」の文字が用いられている表記ゆれの場合が編集距離の範囲が 2 までとした場合に正解が抽出された．よって，編集距離を用いたとき，編集距離が 0 のときにマッチングされなかったアーティスト名が取得できるといえる．

しかしその一方，編集距離 1，編集距離 2 と増えていくごとにアーティスト名と関係ない Wikipedia 記事タイトルと多くマッチングされてしまう．関係ない記事タイトルのマッチングの例を表 8 に示す．例えばアーティスト名が「ショパン」のとき，Wikipedia タイトル名が「あげパン」，「G パン」のように編集距離を用いるマッチングによって，アーティスト名と関係ない記事タイトルとマッチングしてしまう．その解決策として，音楽記事中のアーティスト名と Wikipedia 記事タイトルとマッチングする際に，全記事のタイトル名とマッチングするのではなく，Wikipedia のアーティスト名を含む音楽カテゴリの中の Wikipedia 記事タイトルとマッチングする手法が考えられる．

表 8：アーティスト名と関係ない記事タイトルとのマッチングの例

アーティスト名	Wikipedia タイトル名
ショパン	ショパン
ショパン	あげパン
ショパン	G パン

5.おわりに

本研究では，Web 上の音楽記事に対し，Support Vector Machine を用いてアーティスト名抽出規則を学習することにより，音楽記事中からアーティスト名を抽出する手法を提案した．また，抽出したアーティスト名とアーティストに関する情報へのリンクを取得する手法を提案した．抽出実験において，学習機に与える素性を海外のアーティスト名が出現したときによく使用される中点（・）を与えたとき，再現率，F 値の向上が見られた．よって，中点を素性とすることは有用であるといえる．また編集距離を用いてエンティティリンキングを行うことによって，記事内に表記ゆれが存在していた場合にも正解が抽出できた．

今後の課題としては，再現率をより向上させるために，形態素ごとに判定しているものを，アーティスト名全体を一つにまとめる手法や，リンキングする際に編集距離ではとりきれなかったアーティスト名の略称を含む記事タイトルをマッチングする手法が挙げられる．また，編集距離では多くのアーティスト名とは関係のない記事タイトルとマッチングしてしまうので，その記事数を減らすため，編集距離を用いる際に挿入・削除・置換のコストを 1 として重みづけを行っていたがそれぞれの操作に対して適切な重みづけを行う手法が挙げられる．

参　考　文　献

[1] 古川竜也，相良毅，相澤彰子：言語横断エンティティリンキングのための語義曖昧性解消，情報処理学会誌, Vol.24, No.2, pp.172-177(2014)．

[2] 渡辺一郎，桝井文人，福本淳一：固有表現抽出ツール NExT の精緻化とユーザビリティの向上，言語処理学会第 10 回年次大会 pp.413-415（2004）．

[3] 山田寛康，工藤拓，松本裕治：Support Vector Machines による日本語固有表現抽出，情報処理学会論文誌，Vol.43，No.1，pp.44-53（2002）．

[4] 中野桂吾，平井有三：日本語固有表現抽出における文節情報の利用，情報処理学会論文誌，Vol.45，No.3，pp.934-941（2004）．

[5] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang and Chih-Jen Lin: LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9, pp.1871-1874（2008）．