近代文語論説文を対象とした濁点の自動付与アプリケーション

岡 照晃 （奈良先端科学技術大学院大学）[†1]

Application of Automatic Labeling of *Dakuten* for Near-Modern Literary Style of Japanese

Teruaki Oka （Nara Institute of Science and Technology）

1 はじめに

日本語で初めての大規模均衡コーパスとなる**現代日本語書き言葉均衡コーパス**（**BCCWJ**）[1] が昨年公開された．これにより今，コーパスを利用した日本語研究が急速に増えつつある．

しかし，平安時代や明治時代といった古い時代の資料（**歴史的資料**）のコーパスは，現代語のコーパスほど整備が進んでいない．そのため，日本語研究の大きな位置を占める歴史的研究に，コーパスを用いることは未だ難しい．

歴史コーパスの整備が進まない原因の一つとして，歴史コーパスの整備に必要な**校訂**の作業コストが高いことが挙げられる．コーパス整備の中で必要とされる校訂作業は，歴史的資料の記述中にある**正書法に一致しない表記**をコーパスユーザにとって扱い易い形に修正し，可読性・検索性を上げることである．校訂の作業は現在，すべて人手で行われている．しかし，専門家にしか行えないため，作業人員の確保が大きな課題となっている．また，作業対象が膨大であるため，作業を完了するまでにも時間がかかる．

そこで本研究では，統計的機械学習手法を用い，歴史的資料の校訂作業を自動化することを最終的な目的とする．これにより，誰でも簡単に低コストかつ大規模に校訂作業を実施することが可能になると考えられる．そしてその第 1 段階として，校訂作業の中から**濁点付与**を取り上げ，自動化に取り組んだ [3, 4, 5].

本論文では，文献 [3, 4, 5] で提案した手法を実装した濁点の自動付与アプリケーションの紹介を行う．本アプリケーションはモデルファイルの学習に近代語のコーパス**太陽コーパス** [6] を利用し，対象の文体を近代文語論説文に限定している．しかし，濁点付与に用いるモデルを変更することで，中古和文や近世の資料にも容易に適応可能となっている．

2 校訂作業における濁点付与

歴史的資料の記述中では，図 1 に例示したように，**濁点が付いていることが期待される仮名文字に，濁点が付いていない**ことがよくある．図 1 では例えば,「欲せさる（ホッセザル）」の「さ（ザ）」から濁点が脱落している．本論文では，このような濁点の抜け落ちた文字のことを**濁点無表記文字**と呼ぶ.

表 1 を見ると，明治初期の資料でも濁音の仮名文字の約 83%（368/446）（総文字数の約 4%（368/8,423））が濁点無表記で書かれている．濁点無表記の文字をそのまま残しておくと，歴史コーパスのユーザにとっては読み辛く，検索にも不便である．そのため，濁点無表

[†1] teruaki-o[at]is.naist.jp

今や廣島は其名大に内外國に顯はれ苟も時事を談するものは同地の形勢如何を知らんと欲せさるはあらす是れ征清の大師一たひ海に航せしより　大元帥陛下大纛を此に駐め大本營となし軍務を親裁し玉ふに因てなり先つ其大勢より叙述して次第に細事に及はんとす

（『太陽』 1925 年 2 号 p.64 より抜粋）

図 1: 濁点無表記の例．このテキストは太陽コーパスの原文から抜き出したものである．下線を引いた文字には，通常なら濁点が期待されるが，ここでは付けられていない（濁点無表記になっている）．

表 1: 濁点と濁音の統計．**明六雑誌**[1]第 1 号（2011 年 12 月段階のデータ，総文字数:8,423）中に含まれる濁点文字と濁点を付けることが可能な文字の中から，実際の発音が清音の文字数と濁音の文字数の内訳を調査した．

表記＼発音	濁音（ガ，ザ，ダ，…）	清音（カ，サ，タ，…）	計
濁点文字（が，ざ，だ，…）	78	0	78
濁点を付けることが可能な文字（か，さ，た，…）	368 （濁点無表記文字）	1,273	1,641
計	446	1,273	

記文字を**濁点文字**（濁点付きの文字）に置き換えるという校訂作業が必要になる．この作業を**濁点付与**といい，現在はすべて人手で実施されている．

ただし，濁点付与のような校訂の作業を人手で行うのは非常にコストが高い．特に作業者が専門家に限られてしまうことが問題である．校訂の作業を行うためには，まず対象となる資料を読んで理解しなくてはならない．しかし，歴史的資料の中では現代とは語彙が異なる上，表記や語法の面でも多様であり，読解には専門的な知識が必要である．作業の信頼性を保証するためにも，作業者は歴史的資料の専門家に限定される．だが，そういった専門家を集めることは難しく，作業人員の確保が大きな課題となっている．

作業対象となる資料の量が膨大であることも問題である．例えば，国立国語研究所が構築した太陽コーパスは総文字数約 1,450 万文字の規模[2]である．これに対し，熟練した作業者でも 23 ページ分の資料（約 1 万 6,000 文字）の濁点付与に 1 日掛かりで取り組む必要がある[3]．上述の通り，作業者を大量に確保することは困難であり，人海戦術を取ることができない．そのため，大量の資料を少数人で少しずつ整備していくしかなく，整備を終えるまでに多大な時間が必要となっている．

また濁点付与を人手で行う場合，熟練した作業者であっても表 2 に挙げているような単純なミスをおかすことがよくある．濁点を付与すべき文字を見逃してしまうこともよくある．実際，2 人の作業者がそれぞれに行なった濁点付与の結果を比較してみたところ，濁点を付けた個所の一致率は 84%であり，一致しなかった原因のほとんどが濁点の付け逃しや，表 2 に示したような単純なミスによるものであった．また，コーパス整備の初期段階で 1 人の作業者が実施した濁点付与の結果を完成後のコーパスと見比べた．すると作業者が単独で濁点付与を行なった結果では，濁点無表記文字のおよそ 7%を見逃していたことが分かった．

国立国語研究所では太陽コーパスに引き続き，更なる近代語コーパスの整備が進められて

[1] 1874 年（明治 7 年）～1875 年（明治 8 年）の間に明六社から発行された啓蒙雑誌．全 43 号．
[2] 2005 年 11 月段階のデータ．
[3] 1 日の総作業時間を 5～6 時間とした場合．

表 2: 実際に人手で濁点のアノテーションを行なった場合のミスの例.

ミスのタイプ	正解のアノテーション	実際に作業者が行なったアノテーション
濁点文字に置き換える際の濁点無表記文字の消し忘れ	わが御ためめむぼくなければわたり給事はなし	わが御ためめむぼほくなければはわたり給事はなし
濁点を付与する際の濁点挿入位置の間違い	我が゛外交をして卑屈の平和にあらず゛、	我が゛外交をして卑屈の平和にあら゛す、

図 2: AYTC～にごり ONLY～（ブラウザ内実行時）

いる．しかし，校訂作業の負担から，現場ではその自動化を望む声が上がっている．

そこで本研究では，統計的機械学習の手法を用いて歴史的資料の校訂作業を完全自動化することを最終的な目的とする．これは従来行われてこなかった新しい試みである．歴史的資料の校訂はこれまですべて人手で行われてきたが，作業者不足と作業対象の膨大さから実施に高いコストを必要とした．また人手の作業にミスはつきものである．しかしその作業を計算機に行わせることで，大規模な校訂も低コストで実現でき，単純なミスもなくすことができると考えられる．

自動校訂技術開発の第一段階として，濁点付与の作業を取り上げ，自動化に取り組んだ [3, 4, 5]．最初の目標として濁点付与を取り扱ったのは，濁点付与のタスクが「濁点が抜け落ちた文字に濁点を補う」と明確で取り組みやすかったためである．またタスクが単純な割に，校訂処理の中での必要性は高い．これはコーパスの可読性や検索性向上の観点から，濁点の抜け落ちている問題が無視できないためである．

本論文では，文献 [3, 4, 5] で提案した手法を実装した濁点自動付与アプリケーション **AYTC～にごり ONLY～**（図 2）について述べる．手法の詳細等については文献 [3, 4, 5] に預け，ここではアプリケーションの使い方について述べる．

か，き，く，け，こ，　さ，し，す，せ，そ，　た，ち，つ，て，と， は，ひ，ふ，へ，ほ，　ゝ，／＼（くの字点）

図 3: 濁点付与の対象となる文字.

3 AYTC〜にごり ONLY〜とは

AYTC は，現在開発中の歴史的資料自動校訂ツールの総称である．本論文で述べる**にごり ONLY バージョン**では，先行開発した濁点自動付与モジュール**にごり**を先行実装している．今回の実装では，対象の文体として近代文語論説文のみを扱い，濁点付与の対象となる文字は平仮名と，繰り返し記号である**くの字点**に限っている（図 3）．ただし以下のような処理をアプリケーション内部で行うことによって，明六雑誌のような漢字片仮名交じり文への対応も行なっている．

1. 入力文章中の片仮名文字をすべて平仮名文字に置換.
2. 濁点の自動付与を実行.
3. 濁点を付与した文章の中の平仮名文字を再度すべて片仮名文字に置換し，出力する.

くの字点の表記には，「／＼ [U+3033 U+3035]，／〝＼ [U+3034 U+3035]」と「く [U+3031]，ぐ [U+3032]」のみを認め，「〜〜」や「〜〵」のような表記はくの字点とはみなさない.

本アプリケーション利用の際は，実装手法による濁点付与の精度が 100%ではないことに注意が必要である．濁点付与の性能については 6 章もしくは文献 [3, 4, 5] を参照してほしい.

AYTC〜にごり ONLY〜（以下，単に AYTC）の開発に当たって，多くの文系研究者が本アプリケーションを利用可能なよう，以下の要件を満たすものを目指した.

1. 幅広い PC 環境で動作可能.
2. 利用者が自分の PC にインストールせずとも利用可能．もしくはインストールが簡単.
3. 操作方法が単純明快.

要件 1 と 2 を満たすため，AYTC は Silverlight アプリケーション[4]として開発を行なった．Silverlight アプリケーションはオンライン環境のブラウザ上で動作するアプリケーションであり，アプリケーション自体をユーザの PC にインストールすることなく利用できる．また Silverlight アプリケーションの動作には，Microsoft が提供する Web ブラウザ用プラグイン Silverlight[5]をインストールする必要があるが，Windows OS の PC には購入時からプリインストールされていることもあり，PC の操作に不慣れな者にも敷居は低いと考えられる．プリインストールされていない場合でも，下記のサイトから簡単にインストールすることができる．ブラウザには Microsoft の Internet Explorer だけでなく，Firefox や Chrome，Safari が対応している．また Silverlight アプリケーションは Windows OS だけでなく，Mac OS 上でも動作可能である.

Silverlight アプリケーションは基本的にオンライン環境のブラウザ上で動作する（ブラウザ内実行）．しかし，アプリケーションを各ユーザの PC へインストールすることで，通常

[4]Silverlight のバージョンは 5.
[5]http://www.microsoft.com/ja-jp/silverlight/

図 4: メニューバー．通常，各項目の背景は黒だが，選択時には赤色に塗られる．

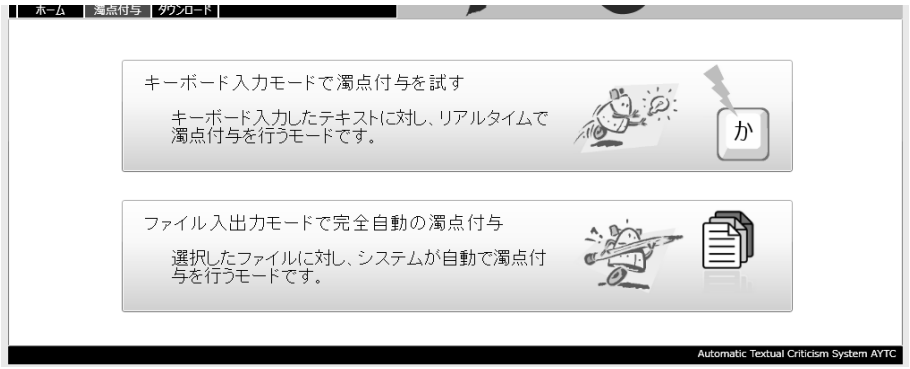

図 5: 濁点付与モード選択画面．

のデスクトップアプリケーションと同様，ブラウザを介さずオフライン環境での動作が可能になる（ブラウザ外実行）．AYTC もブラウザ外実行を念頭に入れて開発を行なっている．インストールには単純な 1 ステップの操作しか必要とせず，面倒な設定等は一切必要ない．

要件 3 を満たすために，AYTC では単純かつ直観的な操作方法を実現している．複雑な操作は一切必要ない．以下の 4 章と 5 章では，AYTC の操作方法について述べる．

4 AYTC による濁点の自動付与

メニューバー（図 4）の項目「濁点付与」をクリックすることで，濁点の自動付与を行うための画面に切り替わる（図 5）．この画面でまず，濁点の自動付与を行うためのモードを選択する．AYTC では，以下の 2 種類のモードで濁点の自動付与を行うことができる．

・キーボード入力モード

・ファイル入出力モード

キーボード入力モードでは，ユーザがキーボードやクリップボードから入力した文章に対し，AYTC がリアルタイムで濁点自動付与を行なっていく．これに対し，ファイル入出力モードでは，指定されたテキストファイルに対して，濁点自動付与が一括で行われる．

以下に，この 2 種類のモードの使い方を述べる．

4.1 キーボード入力モード

図 5 の画面で「キーボード入力モードで濁点付与を試す」をクリックすると，キーボード入力モードの画面（図 6）に切り替わる．このモードでは，入力用テキストボックス（図 6 左）に入力された文章に対し，リアルタイムに濁点自動付与が行われる．濁点付与を行なった文章は出力用テキストボックス（図 6 右）に表示される．この時，濁点を付与した文字は赤字で印字される（デフォルトは黒字）．

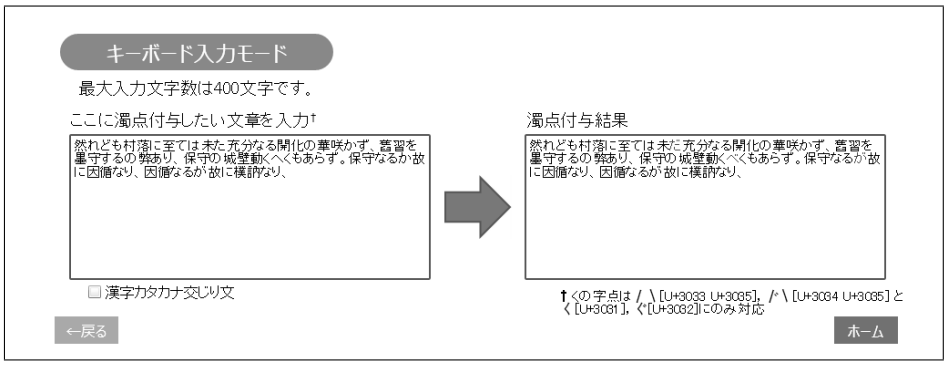

図 6: キーボード入力モード

入力用テキストボックスに入力可能な最大文字数は 400 文字に設定してある．また，入力用テキストボックスの下にあるチェックボックスにチェックを入れることで，漢字片仮名交じり文の入力にも対応する．出力用テキストボックスでは，くの字点が「く [U+3031]，ぐ [U+3032]」に統一表記される．

4.2 ファイル入出力モード

図 5 の画面で「ファイル入出力モードで完全自動の濁点付与」をクリックすると，ファイル入出力モードとなる．ファイル入出力モードでは，以下の順番で入出力ファイルを選択する．

1. 濁点付与を行いたい文章が打ち込まれた「入力ファイル」の選択
2. 濁点付与を行なった後の文章を書き出すための「出力ファイル」の選択

ファイル選択後に表示される「濁点付与」ボタンをクリックすることで，入力ファイルに対する濁点の自動付与結果を出力ファイルに得ることができる．

4.2.1 入力ファイルの選択

図 5 の画面で「ファイル入出力モードで完全自動の濁点付与」をクリックすると，入力ファイルの選択画面（図 7）に切り替わる．この画面ではまず,「参照」ボタンをクリックし，表示されたファイル選択ダイアログから濁点付与を行いたいテキストファイルを指定する．入力ファイルの文字コードは UTF-8 のみに対応しているが，BOM の有無や改行コードには依らない．参照ボタンの左上にあるチェックボックスにチェックを入れることで，漢字片仮名交じり文の入力にも対応する．

入力ファイルにはテキストファイルの他にも，太陽コーパスと同形式の XML ファイルを指定することができる．XML ファイルはファイル識別子.xml で自動認識される．AYTC は XML ファイルの記事・引用タグの属性「文体」で文語文と口語文を区別し，文語文にのみ濁点付与を実行する．

もし，ファイル読み込み時に不具合が生じた場合には，図 8 のようなエラーメッセージが表示される．ファイルの読み込みが問題なく行えると，画面右下の「次へ」ボタンが有効になる．

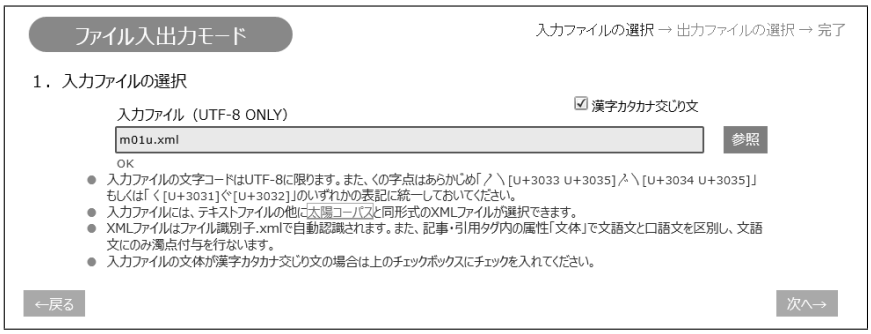

図 7: ファイル入出力モード：入力ファイル選択画面.

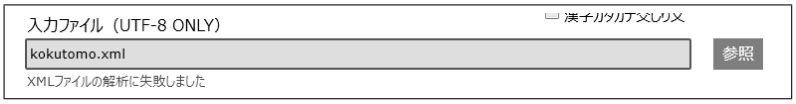

図 8: 入力ファイルの読み込みエラー.

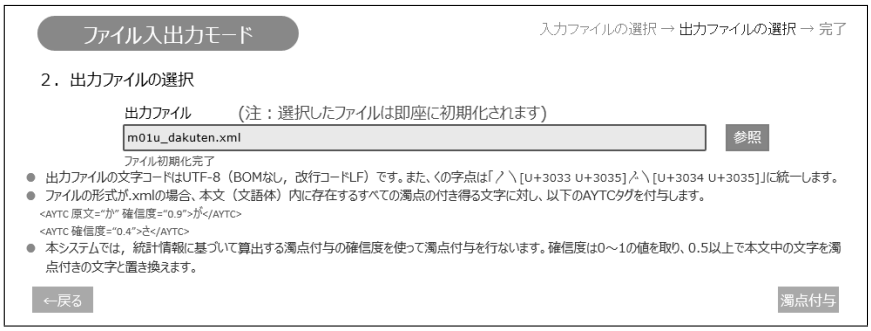

図 9: ファイル入出力モード：出力ファイル選択画面.

4.2.2　出力ファイルの選択

入力ファイル選択画面で「次へ」ボタンをクリックすると，出力ファイル選択画面（図 9）に切り替わる．ここでも前画面と同様,「参照」ボタンをクリックし，表示されたファイル選択ダイアログから，濁点付与結果を書き出したいファイルを指定する．この際，選択されたファイルは直ちに初期化され，空のファイルになってしまうことに注意が必要である．出力ファイルの文字コードは UTF-8（BOM なし，改行コード：LF）で固定となっている．また，くの字点の表記も「／＼ [U+3033 U+3035]，／″ ＼ [U+3034 U+3035]」に統一される．

AYTC には，ファイルのコンバート機能は搭載しておらず，出力ファイルの形式は入力ファイルの形式に準じたものとなる．そのため例えば，テキストファイルでの入力を XML ファイルとして出力することはできない．

ファイルの形式が XML の場合，AYTC は文語記事（もしくは文語引用）の本文中に存在するすべての濁点付与対象文字（図 3）に対し，以下の AYTC タグを付与する．

・濁点付与を行なった場合：〈AYTC 原文=”か” 確信度=”0.9”〉が〈/AYTC〉

– 本文は濁点文字に置き換え，濁点の付いていない元の文字をタグ内の属性「原文」に残す.

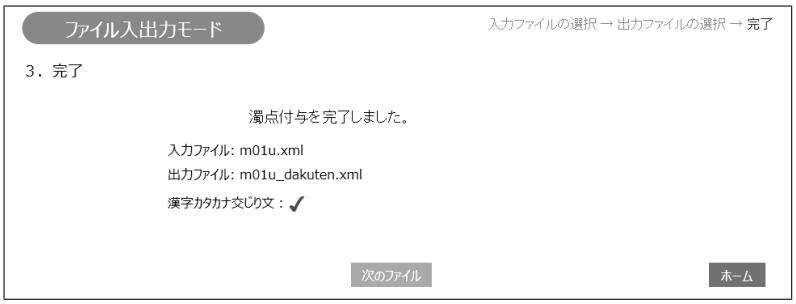

図 10: ファイル入出力モード：濁点付与完了画面.

図 11: インストール画面.

・濁点を付与しなかった場合：〈AYTC 確信度=”0.4”〉か〈/AYTC〉

– AYTC タグを付けるだけで，本文への変更は行わない．また，タグ内に属性「原文」を含まない．

AYTC では，統計情報に基づいて算出する**濁点付与の確信度**を使って各文字に濁点付与を行なっている．確信度は 0〜1 の値を取り，0 に近いほど濁点を付与することへの確信が低く，反対に 1 に近いほど確信が高い事を表している．AYTC では，確信度 0.5 以上で本文中の文字を濁点文字と置き換える処理を行う．各文字に対する確信度は，AYTC タグ内に保持してあるので，タグを見れば各文字がどのくらいの確信をもって濁点を付けたり付けなかったりしたかを確認できる．

出力ファイルを選択後，画面右下の「濁点付与」ボタンが有効になる.「濁点付与」ボタンをクリックすることで，濁点の自動付与が実行され，濁点付与完了画面（図 10）が表示される．

将来的には，この出力ファイルの選択画面と完了画面の間に，濁点付与の対象となっている文章の年代や文体を選択できる画面を挿入する予定である．本アプリケーションが対象とする文体は現在，近代文語論説文だけだが，濁点付与に使うモデル[6]を変更するだけで，中古や近世，近代の口語文の濁点付与にも容易に適応できる．そのため今後のアップデートでは，モデルを自由に選択できる機能を追加することを考えている．

5　AYTC のインストール

メニューバー（図 4）から項目「ダウンロード」をクリックすることでダウンロード用画面へ移動する．このとき，ユーザの PC に AYTC がイントール済みか否かによって，以下の 2

[6]機械学習において,「解決したい問題を数値化する方法」を「モデル」という.

図 12: インストールダイアログ.

表 3: 濁点付与の性能評価.

評価用コーパス	適合率.[%]	再現率.[%]	F 値
SUN-TEST	70.6	96.0	81.4
NF-TEST	95.8	98.0	96.9
M6-TEST	94.5	98.2	96.3

種類の画面のうち一方が表示される.

・ ユーザの PC へ AYTC が未インストールの場合：AYTC のインストール画面（図 11）

・ ユーザの PC へ AYTC がインストール済みの場合：インストール済みを通知する画面

5.1 AYTC のインストール方法

AYTC のインストール画面（図 11）中央の「AYTC をインストール」ボタンをクリックすると，図 12 のようなダイアログが表示される．ここで適宜チェックボックスにチェックを入れ,「OK」ボタンをクリックするだけで，ユーザの PC へ AYTC が自動的にインストールされる.

ブラウザ上で稼働していたアプリケーションがそのままインストールされるので，ブラウザ外実行でも，ブラウザ内実行と全く同じ感覚でアプリケーションを使用できる.

6 AYTC による濁点付与の性能評価

AYTC の濁点付与の性能評価実験を行なった．未校訂の近代文語論説文を対象とし，濁点付与の適合率と再現率を調べた．実験設定及び評価に使用したコーパスは文献 [5] と同じであるため，詳しくはそちらを参照してほしい.

ただし，文献 [5] とは異なり，濁点付与の確信度を 0〜1 の値として得るために，分類器の学習には LIBLINEAR[7]（バージョン 1.8）のロジスティック回帰を使用した [2]．またブラウザ上でも軽快に動作できるよう，学習時に L1 正則化を使い，モデルのパラメータの数を抑えた．[8]

結果を表 3 示す．詳細な結果の考察やエラー分析については，文献 [3, 4, 5] を参照してほしい.

[7] http://www.csie.ntu.edu.tw/ cjlin/liblinear/

[8] モデルは皆パラメータを持ち，一般に，パラメータの数が多いほど精密なモデルであるが，モデルのサイズは大きくなる.

7 おわりに

本論文では，近代文語論説文を対象とした濁点の自動付与アプリケーション AYTC の紹介を行なった．AYTC は Silverlight アプリケーションとして開発を行い，幅広い環境での動作と，簡単かつ直感的な操作を実現している．

今後の課題として，近世の資料や中古和文といった近代文語論説文以外の文体への対応が挙げられる．また，濁点付与以外の校訂作業（e.g., 送り仮名の正規化）の自動化にも今後取り組んでいく予定である．

謝辞

本研究は，国立国語研究所の共同研究プロジェクト「統計と機械学習による日本語史研究」による研究成果の一部である．

文献

[1] Kikuo Maekawa (2008)「Balanced Corpus of Contemporary Written Japanese」In *Proceedings of The 6th Workshop on Asian Language Resources* (ALR 2008), pp.101-102.

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

[3] Teruaki Oka, Mamoru Komachi, Toshinobu Ogiso and Yuji Matsumoto (2011)「Automatic Labeling of Voiced Consonants for Morphological Analysis of Modern Japanese Literature」In *Proceedings of the 5th International Joint Conference of Natural Language Processing* (IJCNLP 2011), pp. 292-300.

[4] 岡照晃, 小町守, 小木曽智信, 松本裕治（2011）「機械学習による近代文語文への濁点の自動付与」情報処理学会研究報告 自然言語処理研究会報告, 2011-NL-201:6, pp. 1-8.

[5] 岡照晃（2012）「統計的機械学習による歴史的資料への濁点の自動付与」第 1 回コーパス日本語学ワークショップ予稿集, pp. 13-22.

[6] 国立国語研究所編（2005）『太陽コーパス』国立国語研究所資料集 15, 博文館新社.