日本語文書校正支援システムの設計と評価†

鈴木 恵美子†† 武田 浩一††

ワードプロセッサが大量に普及し，日本語文書を電子的に作成，配布，印刷することが日常的になってきた．しかし，計算機上でできあがった文書の校正・推敲を行うといった高度のテキスト処理は，最近になってやっと研究が盛んに行われはじめたところで，まだ実用化の段階には至っていない．我々は従来より，機械可読な日本語文書を対象として文書中の誤りや用語の不統一，言い替えた方がよい表現などを検出し，文書の校正支援を行うシステムについて研究してきたが，構造化された文書表現（構造化文書）とその上でのルール形式の校正知識表現を用いることが有効であるという結果を得た．すなわち，1）文書を前処理段階でモデル化することにより，日本語文書のための応用プログラム実行時には字句解析を行うことなく，単語や文節，段落や文書全体といった単位を扱うことができる，2）校正知識は構造化文書上の高レベルの述語として記述できる，3）文書校正知識を複数の段階（入力時と作成時）で利用できるように，対話的文書校正とバッチ的校正が提供できる，といった特長を実現できた．本報告では，我々の開発したシステムとその校正知識，ワードプロセッサを使用した実験により得られた誤りの分類およびその検出可能性について述べる．また，構造化文書の応用として重要語を検出する機能についても検討している．

1. まえがき

現在日本は個人出版の時代とも呼ばれ，計算機やワードプロセッサを使ってユーザが日本語文書を電子的に作成，配布，印刷することが容易になってきた．しかし，計算機によりできあがった文書の校正・推敲を行うといった高度のテキスト処理は最近になってやっと研究が盛んに行われはじめたところで[2),10),17)~19)]，まだ十分な実用化の段階には至っていない．ワードプロセッサの，文書を作成したり印刷したりという基本的な機能は向上しており，ユーザインタフェイスも工夫されているが，英文ではパーソナルコンピュータによるスペル・チェック機能が日常的であるのに対し，日本語ワードプロセッサのユーザはすべての文を自分の目で確かめているのが現状である．

これは主として，日本語の膠着性や日本語文書の正書法が確立していないこと[16)]等が計算機による字句解析，文法処理といった高度のテキスト処理に対する障害となっているためである．したがって英文並のスペル・チェックやスタイル診断[19)]といった校正機能を実現するためには，日本語の性質を考慮した上での校正方法を研究する必要がある．たとえば理想的なワードプロセッサの使い方として，「とりあえず原稿を全部入力してしまい，仮名漢字変換の誤りやミスタイプはあとでまとめてシステムに修正させることにする」という使い方が考えられる．一方では，文書入力時に原稿を見ながら，画面で確認するという方法もある．後者はワードプロセッサの高度化により対応できるが，前者はユーザが入力のリズムを失うことなく後に一括して誤りを訂正できる機能を必要とする．

我々は従来より漢字複合語の短単位分割[13)]や日本語文の文節切り[11)]等の手法を研究しており，これらをもとに，文書を高度に構造化した構造化文書というものを提案した[14)]．この構造化文書では，日本語の文章は文節に区切られ，読みや文法や漢字複合語の短単位構成パターン[13)]などの情報が付加されている．我々は構造化文書上に，「べた書き表現」と「キーワード／文脈表現」というユーザ用の2種類の文書表現と文書校正環境を提供し，対話的に文書を入力すると誤りやスタイル上の問題点を指摘する機構や，作成された文書から校正メッセージと構造化文書を生成するテキスト・コンパイラをもったシステム，CRITAC (CRITiquing using ACcumulated knowledge)[12),22),23)] を実現することを目指している．本論文では，CRITAC の概要とその校正知識について述べ，ワードプロセッサで作成した文書に現れやすい誤りと校正知識の有効性について検討した．また CRITAC の拡張機能についても報告する．

2. 日本語文書の校正技術

最近になっていくつかの日本語文書校正支援システムが試作されているが[1),2),9),10),15),17),18)]，ワードプロセッサの誤りのなかで最も現れやすい仮名漢字変換の誤りや表記のゆれの検出・訂正などについての系統的

† Design and Evaluation of a Japanese Text Proofreading System by EMIKO SUZUKI and KOICHI TAKEDA (IBM Research, Tokyo Research Laboratory).

†† 日本アイ・ビー・エム(株)東京基礎研究所

な手法はまだ得られていない.

石井[1]は，既にできあがった文書をリスト形式に表現し，語用法や文体をチェックしている．そこでは常用漢字表や朝日新聞用語集の知識が Prolog で記述されており，漢字の読み，誤りやすい慣用句，言い替えた方がよいことば使いなどについての情報が出力される.

牛島ら[2]は辞書も文法解析も行わずに文章を字面だけで解析し，推敲するツールを開発している．ここでは句点などに注目して文を切りだし，文頭，文末，文長等を表示したり，字種別の KWIC (Key Word In Context) を作成したり字種別に文字列とレコード番号との相互参照表を作成したりという処理ができるほか，かっこの対応を調べたり，指示代名詞に下線を引いて日本語ラインプリンタに出力させたりすることができる.

絹川[9]は仮名漢字変換レベル，語・句のレベル，文のレベル，段落のレベルの各レベルにおいて，文章の均質化を図ろうとしている．これは複数の人が分担して一つの文書を作成するような場合に，書き方やことば使いの不統一を防ごうとするもので，あいまいさの少ない表現をするために，ワードプロセッサに必要になる機能の検討を行っている.

空閑[10]は文書の作成から校正さらに意志決定という過程を，オフィス業務の一部としての文書作成作業ととりあげて考察している．そして，1）単純な入力ミスを抽出する形態素チェック，2）必須付属語の欠落などを，間違いの多い構文パターンとマッチさせて発見する，3）誤例辞書を用いて同音異義語の誤りを見つける，4）文体の不統一のチェック，5）係り受けのあいまいさのチェック等を行おうとしている.

福島ら[17]は校正だけでなく，論理構造編集，文例提供，文書書き替えなどの機能を提供することにより文書の作成を支援するシステムの構築を目指している.

建石ら[15]は辞書を使わず，構文解析も行わずに，校正の対象とする文書と，不適当な表現を正規表現として集めたファイルとをマッチさせることにより，ユーザに注意を促す方法を提案した．また，一つの文の長さの最大値，平均値等を用いて，文章の読みやすさの測定も行っている.

また安田ら[18]は新聞記事校正等日本文訂正作業の省力化を目指して1）表記のルールに関する誤り，2）一般常識に関する誤りをシステム辞書とユーザ辞書を整備することにより検出しようとしている.

以上日本語文書の校正のために行われている研究のうちいくつかを挙げてみたが，これらは一般的な校正技術や語用法の取扱いについては述べていても，ワードプロセッサの使用によって生じやすい誤変換やミスタイプ等の扱いについてはあまり明らかにしておらず，人手による辞書の整備というような手法を除いてはスタイル以前の誤りを指摘するには至っていない.

2.1 ワードプロセッサで作成された文書の誤り調査

日本語文書中に現れやすい誤りを調査し，分類を行うために，ワードプロセッサで作成された3種類のサンプル文書で誤りを調査してみた*．文書の大きさは，一つ（文書A）が，約9,380文字，もう一方（文書B）が，約 10,480 文字，そして三つ目（文書C）が約 22,000 文字で，3文書とも情報工学分野の論文である．文書Aと文書Bは同一人により入力されたもので，文書Cは別の人の手による．誤りを分類した結果を**表1**に示す.

文書A，文書Bにはワードプロセッサ独特の誤りとも言えるミスタイプや仮名漢字変換の誤りが非常に少なく，ユーザがかなり注意深く文書を入力したと考えられる．一方，その筆者の文章を書くときのスタイルか，句点から句点，読点から読点までが長い文が多いことがわかった．このようにワードプロセッサを '文書清書機' というよりはむしろ，'文書作成機' として使用する場合，ユーザは今まで自分がどのような文章を書いてきたかを確認しながら入力作業を行うため画面に注目することが多い．したがって画面を注視して読みなおしをしている間に仮名漢字変換の誤りやミスタイプに気がついて修正が行われていると考えられる．しかしそれほど注意しているにもかかわらず，ユーザが発見できない誤りがあったのである（例「1

表 1 文書に現れた誤り
Table 1 Preview for types of errors.

	文書A	文書B	文書C
文法的な誤り	1	0	0
仮名漢字変換の誤り	0	0	14
ミスタイプ	0	2	14
おかしなことば使い	2	0	1
一文が長過ぎるもの	2	2	0
句点の打ち方の誤り	1	0	0
長い文節	8	1	0
誤りの数の合計	14	5	29

* 使用したワードプロセッサは IBM 5550 上で動く日本語文書プログラムバージョン4.0である.

入力：二入力」→「一入力：二入力」または「1入力：2入力」のどちらかであるべき).

反面，文書Cでは，文書入力者はブラインド・タイプができるため，原稿のみを見ながらローマ字漢字変換を行っている．その結果ワードプロセッサの画面にあまり注目せず，仮名漢字変換の誤変換を見過ごしたりすることによる誤りが多い(例「文例終←文例集」)．この文書作成者はあらかじめ机上で原稿を書いてしまい，あとで文書を清書するためにワードプロセッサを使っている．これら三つの文書のすべてをとおして，文法的に誤った文は一つしか現れなかった．しかもこれは文書修正時に元の文章の一部を残したまま上から書き加えたことによる，いわば誤修正と考えられ，もともとユーザが文法的におかしな日本語を書いたとは考えにくかった．

この調査をまとめてみると，ワードプロセッサによって作成された文書には，

1. 文法的な誤りの数は意外に少なく，
2. むしろミスタイプや仮名漢字変換の誤りのような局所的に現れる誤りのほうが多い，
3. 長過ぎたり同じ言回しを繰り返し用いたりといったことによる読みにくさが生じやすい，

という性質があると考えられる．

したがって，英文法のように性，数の一致があり，比較的構造のはっきりした文法と異なり，日本語の文法をチェックすることによって，指摘できる誤りは限られているといえ，文書を校正する知識は文法チェックとは異なる独自のものを実際の文書にあたって，経験的，発見的に獲得する手法が有用である．

3. CRITAC

3.1 システム構成

CRITAC のシステム構成は図1のようになっており，大きく分けて次の三つの主要部分：文書前処理部分，ユーザインタフェイス部分，校正用知識ベースからなる．

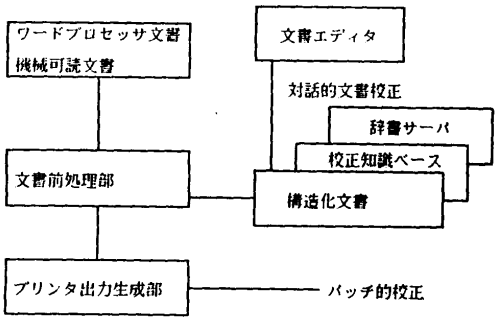

図1 システム構成
Fig. 1 An architecture of text proofreading system.

[文書前処理部分]

文書前処理部分では，べた書きの文書を，

1. 文節切り
2. 自立部と付属部に分離
3. 漢字複合語の短単位分割と読み付け
4. 付属語の接続検定

の順に処理し，図2に示す構造化文書という Prolog の節集合に変える．これにより文書校正のための知識は Prolog の述語として宣言的に表現できる．ワードプロセッサから得られる文書には上記の情報を含むものがあるが，文節区切りや複合語内の語の単位は文書や筆者によりばらつきが生じやすく誤りも多く含むため，ここでは使用せずに前処理部が自動的に行う．

〈文節切り〉

入力された文はまず文節に分けられる．文節切りのためのルールは約 100 個あり，ヒューリスティックな知識をもとに，どのような文字列が現れた際に，どこに文節区切り記号を挿入するかが書かれている．現在，精度は 97.5% である[11]．なお，ここでいう文節は，大河内ら[6]の拡張文節を指す．

〈自立部と付属部に分離〉

文節単位に辞書を引き文節内の自立語を取り出す．

〈漢字複合語の短単位分割〉

取り出された自立語が漢字複合語の場合には，さらにそれを短単位に分ける．日本語はたとえば「不安定」のような語を機械的に短単位に分割する場合，辞書の引き方により，「不安・定」「不・安定」の複数の分割可能性がある場合がある．このようなあいまいさを解消する方法として我々は確率的なアプローチをとることとし，確率付きの漢字短単位辞書を用いている．この方法による漢字短単位分割の精度は約 96.5% である[11]．そして最も確からしい分割が決まった時点

```
校正作業は従来から…
head      tail
head(1, 1, 1,  '校正'. '作業'. nil, 'こうせい'. 'さぎょう'. nil,
               名詞. nil, '12. '12'. nil, 2).
                                          ↑
                                    重要語のレベル
tail(1, 1, 1, 'は'. nil, '係助詞'. nil).
head(1, 1, 2, '従来'. nil, 'じゅうらい'. nil,
               名詞. nil, '12'. nil, 0).
tail(1, 1, 2, 'から'. nil, '格助詞'. nil).
```

図2 構造化文書の例
Fig. 2 Sample structured text.

seg(*I, J, K, X*)	文字列*X*は*I*番目のパラグラフの*J*番目の文の*K*番目の文節である．(以後 *I, J, K* は同じ)
head(*I, J, K, U, Y, G, L*)	*U*は文節中の自立語のリスト．*Y*はその読み，*G*はその品詞のリスト．*L*は*U*が漢字複合語のときその単位構成要素のパターンを表す．
tail(*I, J, K, V, H*)	*V*は文節中の付属語のリスト．*H*は最後の付属語の品詞．
punc(*I, J, K, D*)	この文節に句読点があるとき，*D*がその文字列となる．
sent(*I, J, S*)	*S*は文全体の文字列．
para(*I, P*)	*P*はパラグラフ全体の文字列．
text(*T*)	*T*はテキスト全体の文字列．

図 3 構造化文書の構成要素
Fig. 3 Types of objects in Japanese structured text.

で漢字に読みがふられる．

〈付属語の接続検定〉

付属語オートマトンを用いて付属語の接続検定を行い，正しい付属語列に「五段活用・カ行・連用形」などのカテゴリを表すカテゴリ番号が付けられる．

以上の処理により，入力された日本語文書は**図 3** に示す七つの Prolog の節に変換される．現在，記号，数式，図のような対象を含む文書はうまく扱えない．文書中のこのほかの構成要素はこれらの基本的な節から定義できる．こうして得られた，Prolog の節集合によって表された文書を構造化文書と呼んでいる．

CRITAC ではワードプロセッサで作成されたもとの文書の書式に関する情報を扱わないことにした．これは，書式は文書そのものとは独立に扱うことができ，また出力媒体や文書の用途に応じて容易に指定したり変更したりできるほうがよいと考えたためである．

[ユーザインタフェイス部分]

ユーザインタフェイスは，構造化文書から外部表現を作成して文書を対話的に校正するための文書エディタと，できあがった文書を一括して校正情報や文書処理プログラムへの入力を生成するテキスト・コンパイラ，両者とは独立にオンラインで単語の情報をユーザに提供する辞書サーバからなる[16]．

「べた書き表現（以後ソース表現と呼ぶ）」とは構造化文書の表層の部分だけを抽出したものである（**図 4** 参照）．

「キーワード／文脈表現（以後 KWIC 表現と呼ぶ）」は，構造化文書の各自立語（漢字複合語の場合はその各短単位構成要素）をキーワードとしてその読みの順番等によって並べ換え，前後の文脈とともに表示したものである．したがって構造化文書が *N* 個の自立語／短単位構成要素をもつと，KWIC 表現は *N* 個の行からなる（**図 5** 参照）．

文書エディタではこれら二つの外部表現をとおして，ユーザが対話的に文書を校正できる．現在のところまだ外部表現上の高レベル操作や文書への更新操作

```
CRITAC SOURCE A1  F 72 TRUNC=72 SIZE=211 LINE=138 COL=1 ALT=0
 133 ディタから呼びだされる Prolog のプログラムとして実現されている．
 134
 135 ソース・エディタではソース表現のうえで語単位，文節単位の挿入・削除・更
 136 新が行え，校正機能は表示画面上で多面単位に利用者に文書の誤りを指摘する
 137 という形になっており，校正メッセージを見ながら対話的に文書を変更できる
 138 ．KWIC エディタではキーワードや文脈をその文単位に更新できる．複数の
 139 文にまたがる変更や，文の挿入は KWIC エディタの各行からソース・エディ
 140 タの対応する部分へのスウィッチにより，ソース・エディタ上で行うようにな
 141 っている．
 142
```

図 4 ソース表現
Fig. 4 Source view.

```
CRITAC KWIC A1 F 116 TRUNC=116 SIZE=1138 LINE=520 COL=1 ALT=0
 510    の校正機能は主として仮名   漢字   変換の誤変換と表記のゆ
 511          現在のところは       漢字   の基本語約３００００語
 512    自立語）を中心として仮名   漢字   変換の誤変換や表記のゆ
 513    リスト形式に変換し，常用   漢字表 と朝日新聞の用語調から
 514    よって，指摘できる誤りは   限     られていると考え，文書
 515    書の校正　における技術的   課題   は最近の校正　に関する
 516    い日本語文書の作成は近年   急速   に機械化されるに至った
 517    名漢字変換の誤りのような   局所的 に現れる誤りのほうが多
 518    語文書の作成は近年急速に   機械化 されるに至ったが，英文
 519    ーワードを調べていく校正   規則   が作られている．
 520                 校正   規則   はＰｒｏｌｏｇのルール
 521    あわせて現在２０個の校正   規則   が校正知識ベースとして
 522                 校正   規則   については特に［５］で
 523    作成したり印刷したりする   機能   は向上しつつあるようで
 524    英文では既にパソコン上で   機能   するスペル・チェックさ
 525    たものを含んだ高度の校正   機能   はいまだに商用かされて
 526    よるもので，英文並の校正   機能   を実用かするためには形
 527                 校正   機能   はこのエディタから呼び
 528    ・削除・更新が行え，校正   機能   は表示画面上で多面単位
 529      ＫＷＩＣエディタの校正   機能   は主として仮名漢字変換
```

図 5 KWIC 表現
Fig. 5 KWIC view.

を構造化文書上に反映する機能が実現されていないが，ソース表現上では語単位，文節単位の挿入・削除・更新を行うことを支援する．そして校正機能は表示画面上で，画面単位にユーザに誤りを指摘するようになっていて，校正メッセージを見ながら対話的に文書を更新できるようになっている．KWIC 表現上では，キーワードや文脈を，その文を単位として更新することを考えている．複数の文章にまたがる変更や，文の挿入は KWIC 表現の各行から，ソース表現の対応する部分への切り換えにより，ソース表現上で行われるようにすることができる．KWIC 表現の校正機能は主として漢字仮名混じり語とカタカナ語の表記のゆれの検出である．KWIC 表現上でキーワードを読み順に並べると，同音異義語や表記のゆれをもつ語は必ず隣接して現れる．この特徴を生かして隣り合ったキーワードを調べていくといった校正規則が定義されている．

ユーザインタフェイスの提供する2番目の機能としてテキスト・コンパイラがある．テキスト・コンパイラというのは，構造化文書を入力として文書のソース表現と校正メッセージを出力する．これはちょうど通常のコンパイラがプログラムからエラー・メッセージとオブジェクト・コードを出力するのに対応している．ここでオブジェクト・コードに相当する構造化文書の出力は，

..(単語の区切り)〈単語〉〈読み〉(単語の区切り)..

という形式を考えており，区切り記号や読みの有無はオプションで指定できる．このような出力はたとえば日本語文書の音声化，点字化や検索システムへの入力として利用できる．テキスト・コンパイラではさらに文書の統計的情報等の付加情報を合わせて出力できるようになっている．付加情報の種類としては文献2)で考察しているようなものが有用であると考えている．テキスト・コンパイラの出力結果を**図6**に示す．ユーザはこの出力を見ることにより，誤変換を見つけたり，言回しを変えた方がよいような表現について，推敲したりできる．

辞書サーバは，ユーザがエディタから対話的に各種の辞書情報を検索できるように，関係データベースシステム SQL/DS[21]の関係として国語辞書を管理したものである．現在のところ漢字の基本語約 30,000 語が格納されており，正書，読み，品詞の三つの属性をもっている（**図7参照**）．ユーザは校正機能により指摘された漢字複合語の誤りをこの辞書を用いて確認したりできる．さらにたとえば『意味分類』という関係を追加すれば，言回しを変えたいときに辞書から同義語を検索することができる．また構造化文書は容易に関係データベースとして格納できるため，完成した文書を文書データベースとして辞書と同様に管理することにより，オンラインの単語の用例検索や一般的な文

```
318
319 * ERROR 14 on line 49 position 32: 『即ち』
320       文が等位接続詞で始まっています.
321
322 * ERROR 31 on line 23 position 6: 『機械翻訳システム野ためには』
323       適当でない漢字が使われています.
324       誤変換ではありませんか？
325
326 * ERROR 33 on line 40 position 35: 『実用かと』
327       未変換の可能性はありませんか？
328       この「か」は漢字で書かれるはずではありませんか？
329
330 * ERROR 17 on line 24 position 28: 『よって』
331       一つの文の中に同じことばが繰り返し使われています.
332       他の言回しができませんか？
333
334 * ERROR 14 on line 42 position 21: 『即ち』
335       文が等位接続詞で始まっています.
336
337 * ERROR 19 on line 38 position 20: 『できあがった』
338       おかしなことば使いをしています.
```

図6 コンパイラ出力
Fig. 6 Sample proofreading messages.

```
CRITAC   KWIC     A1  F 136  TRUNC=136 SIZE=232 LINE=55 COL=1 ALT=0
+-------------------------------------------+
| 更生 :   こうせい            :000820      |
| 公正 :   こうせい            :000060      |
| 硬性 :   こうせい            :000020      |
| 恒星 :   こうせい            :000020      |
| 抗生 :   こうせい            :000020      |
| 構成 :   こうせい            :000020      |
| 攻勢 :   こうせい            :000020      |
| 後世 :   こうせい            :000020      |
| 校正 :   こうせい            :000820      |
| 鋼製 :   こうせい            :000020      |
| 厚生 :   こうせい            :000020      |
| 高声 :   こうせい            :000020      |
+-------------------------------------------+
 55  しかし出来あがった文書を      校正
 56      本報告では日本語文書      校正
 57  の短単位分割や日本語文の      校正
 58  って表現された構造化文書      校正
 59  って表現された構造化文書      校正
 60  を支援するための試作シス      校正

・推敲したりする補助手
を支援するための試作シ
環境を実現した。
環境を実現するものと考
知識、誤変換や表記のゆ
について述べた。
```

図7 同音語の出力 (SQL/DS)
Fig. 7 Homonyms returned from the dictionary server.

書検索のように広い範囲の文書情報の要求や文書の蓄積による大容量化にも対応できる．辞書サーバへのアクセスは，あらかじめ用意された同音語等の検索以外にも，質問言語 SQL を用いてユーザが動的に表現できる．これは通常の電子辞書と違う関係データベース・システムのもつ大きな利点である．また，辞書の属性を増やすことにより，さらに文書校正上有効な情報をユーザに提供することが可能となる．

[校正用知識ベース]

CRITAC の校正知識はソース表現上で使用されるものと，KWIC 表現上で使用されるものの2種類に大別される．前者をソース表現上のルール，後者をKWIC 表現上のルールと呼ぶ．以下では，この校正知識について詳しく述べる．

3.2 CRITAC の校正知識

CIRTAC の校正知識を用いて説明する．

[ソース表現上のルール]

ソース表現上のルールは，複雑であいまいな名詞句に警告を与えたり，ミスタイプや同じ表現の繰り返しを見つけたりする．日本語の文法はある種の助詞を繰り返し用いて一つの長い名詞句を作ることを許してしまうため，一度読んだだけではわからないような，係り受けのあいまいな表現がある．文章中のそのような個所に警告を与えてユーザに書き替えを促したり，辞書に載っていないような語が現れたときにミスタイプの可能性はないかユーザに尋ねたりするのである．ここでは不均一な語の使用を見つけて警告するルールを示す．

〈例1〉ルール番号 106：不均一な語の使用

XとYはテキスト中に現れる2種類の短単位構成要素とする．ここで，ある漢字複合語 XY があり，同時にテキスト中に表現 XCY が現れていたとする．Cは任意の表現である．

CASE 1：Cが漢字接尾語の場合 → 警告する

CASE 2：Cがひらがなの場合

CASE 2-1：Cが等位接続詞の場合 → 何もしない

CASE 2-2：上記以外の付属語の場合 → 警告する

CASE 3：上記以外の場合 → 何もしない

この規則により，たとえば文書中のあるパラグラフでは「編集画面」と使っている語が別の場所で「編集用画面」と使われていた場合，「用」は漢字接尾語なのでCASE 1 により警告を受ける．また同じく，「編集のための画面」も CASE 2-2 により警告される．しかし，Cとして「および」「または」などの語がくる場合，たとえば「編集および画面」「編集または画面」という句は「編集画面」とは全く異なる概念を指している．この場合 XY は同じ重みをもつ対等な語として扱われていると考えられるため，警告はしない．

次にもう一つ，誤変換の例について見てみる．

〈例2〉ルール番号 304：助詞の誤変換

X，C，Y はテキスト中にこの順に現れた3種類の短単位構成要素とする．ここでCはある助詞と読みを同じくする漢字である．

CASE 1：Yが読点の場合→警告する

CASE 2：Yがひらがなの場合

CASE 2-1：Yが文法的にCに接続できない場合 → 警告する

CASE 2-2：上記以外の場合 → 何もしない

CASE 3：Yが漢字短単位構成要素またはカタカナ語，あるいはアルファベットで書かれている場合

CASE 3-1：Xが漢字でない場合 → 警告する

CASE 3-2：Xが漢字でYがアルファベットまたはカタカナで書かれている場合 → 警告する

CASE 3-3：上記以外の場合 → 何もしない

CASE 4：上記以外の場合 → 何もしない

このルール304は，本来ひらがなのままでよい助詞が誤った操作により漢字に変換された場合を想定して，それをチェックするためのルールである．今，Cが漢字の「野」であったとする．続く文字が「，(読点)」であった場合（「知識工学分野，人工知能分野，…」のように「分野」という語の一部が「，」によって繰り返されている可能性もあるが，ここではCは一つの独立した短単位構成要素と仮定しているので，そのような場合はない．），「野」は単独に一語で現れてそこで「，」によって中止されていることから，「の」あるいは「や」が誤って変換されたのではないかと考えられる．そこで CASE 1 に従って警告する．また「野」の次に「ため」のようなひらがな語が現れた場合は「野」が単純な名詞であるとすると，接続不可能なので同じく警告する．それ以外のたとえば「から」「にも」などの助詞の場合は，「野」が名詞であるとすると「野にも山にも…」「野から出てきて…」等の使い方もあることから，これについては何もしない．CASE 3 の場合はYが漢字短単位構成要素またはカタカナで書かれた語またはアルファベットで書かれた語であるから，Xが漢字でないとき，つまりカタカナかアルファ

ベットかひらがなで書かれているときにはCの直前に単語の区切りがある可能性が高い．そして特にYがカタカナもしくはアルファベットで書かれている場合には，「野」で始まる外来語とも考えられ，不自然である．たとえYが漢字短単位構成要素であるとしても，「野」が接頭辞的に働くことになり，これも不自然なので警告する．Xが漢字であるとしても，Yがアルファベットもしくはカタカナの場合は，CはXとYを結ぶ助詞的役割を果たすことが多いはずなので警告する．ソース表現上での校正ルールの実行の様子を図8に示す．

[KWIC 表現上のルール]

KWIC 表現上のルールは，正書や読み等で順序付けられたキーワードに対する述語である．日本語文書によく現れる，誤変換や表記のゆれのほとんどは，適当なキーワードの順序を与えることにより，KWIC 表現のうえで隣接したキーワード間の関係としてとらえることができる．たとえば，例1のソース表現上のルールは，キーワードが正書の順に並んだ KWIC 表現のもとで次のように表せる．

〈例3〉 KWIC 表現のルール

```
CRITAC  SOURCE  A1  F 72  TRUNC=72 SIZE=211 LINE=120 COL=1 ALT=0
115
116 によって［5］の構造化文書というＰｒｏｌｏｇの節集合に変換する。もとの
117 文書のなかで書式に関する情報は現在ＣＲＩＴＡＣでは扱っていない。書式は
118 文書そのものとは独立に扱え、また出力媒体や文書の用途に応じて容易に指定
                                                          32==
119 ・変更出来るべきであると考えている。この時点でＣＲＩＴＡＣで扱う前処理
    ========================                              36====
120 語の文書は英文並の区切られた単語の列であるよい。ただし、この文書には書
    ====          15========
121 く自立語の読み、自立語や付属語の瀕死などといった情報が埋めこまれている
              24====
122 。
123
124 利用者インターフェイスでは構造化文書から図２のような外部表現を作り出し
```

図 8 ソース表現上での校正ルール適用画面
Fig. 8 Source view with errors underlined.

```
CRITAC  KWIC    A1  F 136  TRUNC=136 SIZE=832 LINE=675 COL=1 ALT=2
+-----------------------------------------------------------------------+
| 表記のゆれの可能性があります。                                        |
+-----------------------------------------------------------------------+
668 節から導出できる文書構成    要素等      をＰｒｏｌｏｇで記述し
669 井：計算機による日本語の    用語        ・固有名詞の校正、ＩＣ
670 良してこれらを扱うことが    予想        される。
671 Ｃの知識ベースに追加する    予定        である。
672   現在このような拡張性を    より        高めるために、構造化文書
673           ［プログラム    呼び出し    ］  構造化文書に対して利
674 部表現や校正知識ベースの    呼び出し    による校正メッセージの生
675 ｒｏｌｏｇのプログラムの    呼出し      が指定できる。
676 ｇで記述しており、これを    ライブラリ  としてＣＲＩＴＡＣの知識
677 どを大まかに把握するのに    利用        できる。
678 対話的校正を行なったり、    利用者      がＰｒｏｌｏｇで記述した
679 し］  構造化文書に対して    利用者      が特別に実行したいＰｒ
680 関するものは、文書入力の    リズム      を乱すことや、チェックに
681           これにより、    例          えば人名のような特定の語
682             前者の        例          としては、ＥＰＩＳＴＬＥ
683 切り）・・・という可変長    レコード    形式のファイルを指定で
684 文頭、文末のパターンや、    論旨        の流れなどを大まかに把握
```

図 9 KWIC 表現上での校正ルール適用画面
Fig. 9 KWIC view with error on the line No. 674 and line No. 675.

今正書の順に並べられた KWIC 表現の第 i 行のキーワードを K(i), K(i) に後接する付属語連鎖を Kf(i), そのあとに現れる自立語を Kj(i) と書く．Kf(i) は『による』や『の』といった単純なものか，接続詞『および』等からなるものとする．直観的には，例1の X, Y, C はそれぞれ K(i) Kj(i), Kf(i) に対応する．

CASE 1：ある接尾語Cがあり，K(i+1)=K(i)C, Kf(i) と Kf(i+1) は空，および Kj(i)=Kj(i+1) が成り立つ．→警告する

CASE 2：K(i)=K(i+1), Kf(i) が空，Kj(i)=Kj(i+1) で，Kf(i+1) が空でない場合

CASE 2-1：Kf(i+1) が接続詞の場合→何もしない

CASE 2-2：上記以外の場合→何もしない

CASE 3：上記以外の場合→何もしない

KWIC 規則のもう一つの利点は，誤りが隣接したキーワードの上で表示できるため，人間にとって非常に理解しやすいことである．例1のソース表現の規則では，文書に点在した誤りを検出しても，それをわかりやすく示すのは容易でない．

同様にして KWIC 表現上で「呼び出し」と「呼出し」といった表記のゆれを検出する規則や同音異義語を検出する規則を得ることができる．この場合はキーワードが読みの順に並べられなければならない．図9に読み順に並んだ KWIC 表現で，「呼び出し」と「呼出し」の表記のゆれを検出した例を示す．このように，KWIC 表現ではキーワードの順序が本質的である．より複雑な順序として，キーワードのみでなく，それに前接する語や後接する語の正書や読みを組み合わせることにより，かなりの範囲で誤りを指摘できると考えている．

現在のところ校正ルールの数は 30 個で，種類は表2に示すとおりである．

4. 実験結果・評価

4.1 誤りの分類と調査

我々はワードプロセッサで作成された2種類の比較

表 2 ルールの種類とカテゴリ
Table 2 Categories of rules.

カテゴリ	番号	ルールの説明
スタイル	101	文の長さ
	102	自立語の長さ
	103	付属語の長さ
	104	接続詞の使用
	105	受身の使用
	106	表記のゆれ
	107	助詞の使い方
	108	漢数字とアラビア数字
	109	かなとアルファベット
	110	ことば使い
	111	慣用句の使い方
ミスタイプ	201, 203	句点の使い方
	204	読点の使い方
	205	ローマ字かな入力
	206	ひらがな列
	207	かっこの対応
誤変換	301	同音異義語
	303	未変換の漢字接尾語
	304	助詞の誤変換
	305	接続検定
	306	漢字複合語の中の誤り

的短いサンプル文書でそこに現れる誤りを，3.2 節で述べた校正ルールがどの程度検出することができるかを調査してみた．この際，あまり作成中の画面に注目せず，下書きの原稿を見ながら，無造作に変換キーや文字種キーを押した．なお，ここではローマ字かな変換入力を行い，文節単位の仮名漢字変換を行った．文書の大きさとしては，一つ（文書S）が 2,099 文字，もう一方（文書T）は 2,779 文字，2 文書とも情報工学分野の論文である．文書Sも文書Tも筆者のうちの1人が入力したもので，文書Sはしばらく時間をおいてから，同じワードプロセッサを使用して今度は比較的注意深く入力してみた結果文書 S′ を得た．誤りを分類した結果を**表 3** に示す．

ここで，誤変換に 2 種類あるのは，誤変換 α というのが，いわゆる同音異義語による変換誤りで，誤変換 β というのが，ミスタイプによる誤変換である．

未変換についても誤変換と同じく，未変換 α は単純に変換キーの押し忘れと思われるもので，未変換 β はミスタイプのせいで，該当する漢字が見つからず，変換されずに残ってしまったと思われるものである．

その他のミスタイプには，熟練者には起こらないであろうが，「でけあがった←できあがった」（本当は右手中指の「i」を打とうと思っているのに，左手中指の「e」を打ったことによる誤り）や「コーコ←コード」（本当は左手中指の「d」を打とうと思っているのに，右手中指の「k」を打ったことによる誤り）のような，右手と左手の動作誤りが特徴的であった．

表 3 サンプル文書に現れた誤り
Table 3 Types of errors appearing in 3 documents.

	文書S	文書T	文書 S′
誤変換 α	24	20	3
誤変換 β	2	0	1
未変換 α	8	0	0
未変換 β	1	2	0
ミスタイプ	7	5	2
文字種キーの押し忘れ	2	1	0
変換され過ぎ	10	5	0
固有名詞	2	0	0
表記のゆれ	0	4	0
スタイルの乱れ	0	1	0
誤修正	0	1	0
誤りの数の合計	56	39	6

文字種キーの押し忘れによる誤りは思いのほか少なく，全体で 3 種しか現れなかったが，これらの内の二つはカタカナ語のあとでひらがなキーを押さずに続けて付属語を入力してしまっており，残る一つは，カタカナ語（プログラム）がひらがなで書かれていた．『変換され過ぎ』というのは，長い単位で変換したときや，または助詞がカタカナ語とカタカナ語をはさんで途中にあるときに変換キーによって変換されてしまったような場合を指す．

表記のゆれは，外来語，専門用語に多く，アルファベットで書かれる場合にもカタカナで書かれる場合にも起こりやすい（例「PROLOG」と「Prolog」，「インターフェース」と「インターフェイス」など）．

誤修正はもともと正しく書かれていたであろう語の一部が消去されてしまっていたものである（例「辞サーバ」←「辞書サーバ」）．

4.2 実験結果と評価

KWIC 表現上で単語を読みの順に並べ，読みが同じで表記の異なる語が現れていたときに警告するようなルールを働かせることで，誤変換の内，現在 48.9% は検出できる．誤変換 β については「救って←作って」や「苦情←向上」など，この文節が本当に誤りかどうかを判定するには品詞や語の使用頻度だけでなく，文の意味まで考えなければならない．これらの文節がローマ字かな入力で，「tsukutte」と打つべきところを「t」を打ち損じたかあるいは何らかの操作ミスにより消してしまったかで「sukutte」となってしまったとか，同じように「kouj(y)ou」の最初の「o」が抜

けて「kuj(y)ou」になったのであろうといった細部について想像はできても[7]，英語などのように，入力した文字がそのまま出力される言語と異なり，日本語ワードプロセッサでは，一番多くの場合，日本語の読みからローマ字つづり，ローマ字からかな，かなから漢字へと3段階の変換が行われており，最終的な漢字の出力から本来入力されるべき文字列を想像することはかなり難しい．CRITAC では現在のところ，構文的な情報や意味は扱わないので，誤変換 β に対しては検出・訂正の方法がない．

未変換 α については，変換されるべき正しいかな文字列が入力されているが，構造化文書では正しく自立語と付属語に分離できない可能性がある．今回のサンプルでは8個の内の1個については正しく文節切りできない．残り7個については文節切りは正しいので，文節内の構造をうまくとらえられれば辞書を引くことでひらがな書きの未変換語を漢字に変換できる．一方未変換 β の内文書Tに現れた「ひょうげんん←表現」についてはひらがな「ん」が二つ続くような語はないので警告できる．しかし文書Sに現れた「ねべてきた←述べてきた」については，たとえばミスタイプは一つの文節中では1か所にしか現れず，かつ文書の入力が常にローマ字仮名漢字変換であるとすると，そのローマ字表現「nebetekita」の内のどこか1文字を別のアルファベットで置き換えることで，語を創造できるが，そこから派生する語は非常に多くなる可能性があり，処理も複雑なので，当面はこの種の誤りの訂正は考えない．ただし，「ねべる」という語は辞書になく，文節区切りも失敗するので検出だけは可能である．

文字種キーの押し忘れによる誤りは CRITAC では，付属語接続検定の折に接続に失敗し，かつそのような自立語（ワード・プロセッサニオイテ，チェックサエナク）も辞書にないことから，検出し警告することができる．

以上の考察から，現在どの程度の誤りが検出できるか，将来はどの程度まで可能であるかについて，誤りの種類別にグラフにしてみた（**図10**参照）．

5. CRITAC の拡張機能

最近，新聞記事を入力として意味解析，要約処理をし，記事の要約リストを出力する要約支援システム[8]のようなものも考えられはじめている．そこで，CRITAC でも構造化文書という文書モデルを有効に活用する一つの応用として重要語検出機能を追加した．

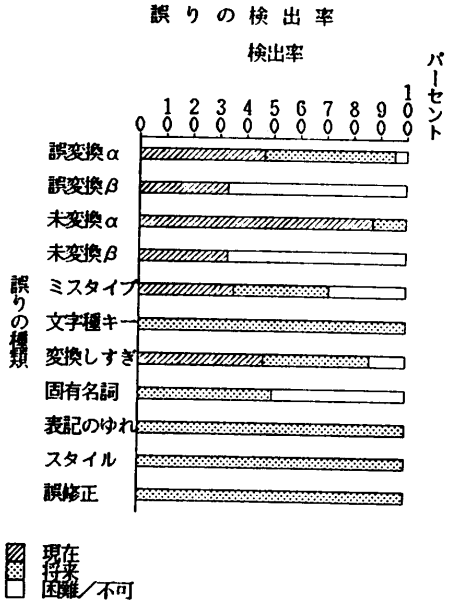

図 10 誤りの検出・訂正率
Fig. 10 Detection or correction rate of mistakes.

欧米の文書では，単語の頻度情報をもとに，重要語を抽出する手法が提案されている[20]．表意文字である漢字を欧米語の単語に対応させれば，漢字の頻度情報をもとに同様の処理で重要語を自動抽出する手法が考えられる．我々は，漢字の出現頻度や，漢字カタカナ列の頻度情報に注目して，語の分類や重要語の抽出を機械的に行える梅田ら[3)~5)]の手法を利用した．この機能を実現するためには CRITAC で本来利用していた構造化文書という文書表現の中に，自立語が重要語になりうるかどうかの情報を付け加えるだけで済む（図2参照）．

5.1 重要語検出法

重要語を検出するためのキーは，漢字もしくはカタカナ列とする．トレーニングデータとして，JICST の論文抄録を用い，JICST の文献分類カテゴリに対応して出現頻度をカウントする．その後，分類カテゴリ間の頻度分布を計算し，分布に偏りがあるものだけを，重要語となる漢字もしくはカタカナ列とするのである．類似した偏りのあるもの同士をまとめるには，クラスタ分析を応用した手法を用いた．文中の名詞列を対象に次のいずれかの条件を満足するものを，重要語とする．

1）キーとなる漢字を二つ以上含み，かつ文字数が3文字以上の名詞列
2）キーとなる漢字を一つ含み，かつ文字数が4文字

以上の名詞列

3）キーとなる漢字を一つ含み，かつカタカナ列を含む名詞列

その結果，多少ノイズとなるものも抽出されるが，「もれ」を少なくすることを優先して考慮した．

5.2 拡張機能の動作

重要語検出機能は，CRITAC の2種類のモード(対話的モードとバッチ的モード）で使用できる．

[対話モード] 対話モードでは，ユーザはソース表現と KWIC 表現という2種類の表現を提供されている．重要語検出機能はソース表現上では重要語に下線が付され，かつ重要語のレベル（重要度）が表示される（**図11**参照）．そして KWIC 表現上ではたとえば重要度5以上の語をリストにして表示するなどの操作が可能である．重要語のレベルを付けるには各種の方法が考えられるが，ここでは重要語のもつ生起確立 10^{-x} の指数値 x を用いた．この機能を使うことにより，ユーザは自分がどの段落でどのような語（重要語）を用いて論を展開したか，その章のレベルと内容が適切かどうかなどについて確認できる．

[バッチモード] バッチモードはもともと段落や文書全体といった大きな単位での校正処理や各種の統計情報を利用するきめの細かい文書の校正を目的としている．重要語検出機能はその目的をさらに追及するために有効である．

たとえば，

出現個所	重要語	回数
1章　段落1	校正作業	5
	日本語	3
	辞書ベース	2
1章　段落2	ワードプロセッサ	3
	ユーザインタフェイス	2
2章　段落1	文書データベース	2
	漢字複合語	2
	統計情報	2

というような重要語情報が示された場合，その章の内容はある程度推測することができる．また，

初出個所	重要語	付属語	文末表現
段落1　行5	校正作業	とは	である
段落2　行2	校正知識	は	用いる
段落2　行12	文書データベース	を	定義する

のような出力情報は，重要語の初出個所とその文章の性質を予測できる．重要語は最初に出現する際，誤解をまねかないためには，その語の定義を与えるべきであるといった経験則から，定義文でない可能性のある

「校正知識は…用いる」

の文を取り出し，本当に校正知識の定義を与えていないかどうか，実際の文章にあたって確認するようユーザに促すこともできる．

6. まとめ

本稿ではワードプロセッサによって作成された文書に現れやすい誤りを指摘するため，まず実際の文書に現れる誤りを調査，分類した．さらにそれらの誤りの内どの程度が機械的に検出可能かについて検討した．また，より高度な文書処理機能の一例として重要語検出を考え，試作システム上に実現した．

本研究の一部はワードプロセッサの高機能化によって代替できるものもある．しかし，文書中に出現する同じ読みをもつ語を一度に見せる機能 (KWIC) など，文書作成中というよりは作成後に働かせる機能もあり，本稿では合わせて文書の校正ということで検討した．

謝辞 本研究を遂行するにあたり，日本アイ・ビー・エム株式会社の同僚，先輩から助力や助言を得たことに謝意を表します．特に校正ルールのインプリメンテーションおよび討議に加わって頂いた西野哲朗氏，沼尾雅之氏，丸山宏氏，尾関正俊氏そして梅田茂樹氏に感謝しま

```
CRITAC   SOURCE   A1  F 80  Trunc=80 Size=57 Line=23 Col=1 Alt=1
  14 て、文書校正のツールにより、すでに出来上がった文書に対して語用法や文体
                (1)-----                                        (1)---
  15 チェックのためのプログラムや、文章の質を評価することにより、より良い文
                          (1)-------
  16 章に改良する方式などが試作検討されつつある。
  17
  18 我々は従来から漢字複合語の短単位分割や日本語文の文節切り等の手法を研究
                                (1)-------  (1)-----  (1)-----
  19 仕手折、これらの技術的基板から構造化文書をいう文書の概念構造を提案し、
                                      (1)-------        (1)-----
  20 このうえで利用者用の二つの外部表現および文書校正環境を実現した。本報告
                      (1)------      (1)-----      (1)----------
  21 ではこれらの概念とその試作システム野うえで実現した機能について述べる。
                                (1)-----------
  22 文書前処理部ではべた書きの文書を構造化文書というプロローグの節に変換す
     (1)----------                    (1)--------
  23 る。もとの文書のなかで書式に関する情報は現在は取り扱っていない。書式は
                                (1)-                            (1)-
  24
```

図 11 ソース表現上での重要語表示画面
Fig. 11 Source view with keywords underlined.

す．さらに，本研究の機会を与えてくださった藤崎哲之助氏に深謝します．

参考文献

1) 石井：日本語の漢字・用字の校正のための知識，Institute for New Generation Computer Technology, Technical Report : TR-0092 (1985).
2) 牛島，日並，尹，高木：日本語文章推敲支援ツール「推敲」のプロトタイピング，コンピュータソフトウェア，Vol. 3, No. 1, pp. 35-46 (1986).
3) 梅田，諸橋，細野，後藤，綾部，原田：漢字カタカナ列の頻度情報に基づいた日本語文献の自動分類，第 32 回情報処理学会全国大会論文集，4 T-10, pp. 1687-1688 (1986).
4) 梅田：漢字の出現頻度特性を用いた日本語文献の機械処理，情報管理，Vol. 29, No. 5, pp. 410-420 (1986).
5) 梅田，諸橋，細野，原田，後藤：漢字クラスターによる日本語文献の重要語抽出，情報処理学会自然言語処理研究会資料，58-5 (1986).
6) 大河内：仮名漢字変換のための形態素接続規則，日本アイ・ビー・エム(株)東京サイエンティフィック・センター・レポート，N : G 318-1560-1, 19281 (1981).
7) 角田：多層テキスト構造を持つ日本語エディタ，第 27 回プログラミングシンポジウム，pp. 75-84 (1986).
8) 北，小松，安原：要約支援システム COGITO，情報処理学会自然言語処理研究会資料，58-7 (1986).
9) 絹川：高品質日本語文章作成支援機能の一考察，第 31 回情報処理学会全国大会論文集，4 H-6, pp. 1389-1390 (1985).
10) 空閑：文書作成・校正支援システム WISE，電子通信学会技術報告，OS 86-28, pp. 13-18 (1986).
11) 鈴木：漢字かな混じり文に現われるひらがな列の文節推定方法について，第 31 回情報処理学会全国大会論文集，4 H-3, pp. 1383-1384 (1985).
12) 鈴木，武田，藤崎：日本語文書校正支援システム CRITAC，情報処理学会文書処理研究会資料，8-5 (1986).
13) 武田，藤崎：統計的手法を用いた漢字複合語の短単位分割，自然言語処理研究会資料，48-2 (1985).
14) 武田，鈴木，西野，藤崎：日本語文書作成支援システム CRITAC の校正知識，第 32 回情報処理学会全国大会論文集，4 T-13, pp. 1693-1694 (1986).
15) 建石，小野，山田：Diction と Style の日本語化について，第 31 回情報処理学会全国大会論文集，4 H-2, pp. 1381-1382 (1985).
16) 長尾（監修）：日本語情報処理，電子通信学会編 (1984).
17) 福島，大竹，大山，首藤：日本語文章作成支援システム COMET，電子通信学会技術報告，OS 86-21, pp. 15-22 (1986).
18) 安田，島崎，高木，池原：日本文訂正支援システム REVISE，第 33 回情報処理学会全国大会論文集，4 J-9, pp. 1719-1720 (1986).
19) Cherry, L. L. : Writing Tools, *IEEE Trans. Communication*, Vol. COM-30, No. 1, pp. 100-105 (1982).
20) Salton, G., Yang, C. S. and Yu, C. T. : A Theory of Term Importance in Automatic Text Analysis, *J. Am. Soc. Inf. Sci.*, Vol. 25, No. 1, pp. 33-44 (1975).
21) IBM Corp. : SQL/Data System Concepts and Facilities, GH 24-5013 (1983).
22) Takeda, K., Suzuki, E., Nishino, T. and Fujisaki, T. : CRITAC—An Experimental System for Japanese Test Proofreading, *IBM J. Res. Dev.*, Vol. 32, No. 2, pp. 201-216 (1988).
23) Takeda, K., Suzuki, E. and Fujisaki, T. : A User Interface of a Text Proofreading System, *Proc. IEEE Symposium on Office Automation*, pp. 15-24 (1987).

（昭和 63 年 12 月 8 日受付）
（平成元年 9 月 12 日採録）

鈴木恵美子（正会員）

昭和 33 年生．昭和 56 年筑波大学第三学群情報学類卒業．昭和 58 年同大学院修士課程修了．同年日本アイ・ビー・エム(株)に入社．同社東京基礎研究所にて統計的日本語処理，文書校正システムなどの研究を行い，現在日英機械翻訳の研究に従事．計量国語学会会員．

武田　浩一（正会員）

昭和 33 年生．昭和 56 年京都大学工学部情報工学科卒業．昭和 58 年同大学院修士課程修了．同年日本アイ・ビー・エム(株)に入社．同社東京基礎研究所にて統計的日本語処理，文書校正システムなどの研究を行い，現在英日機械翻訳の研究に従事．昭和 62 年～平成元年までカーネギーメロン大学機械翻訳センター客員研究員．電子情報通信学会，日本ソフトウェア科学会，ACM 各会員．