自然言語処理 120－21
(1997. 7. 25)

知識ベースに基づく点字翻訳のための日本語文節区切り手法

鈴木 恵美子† 小野 智司‡ 平岡 大樹‡ 狩野 均‡ 西原 清一‡

†東京家政学院筑波女子大学短期大学部 情報処理科 ‡筑波大学 電子・情報工学系
〒305 つくば市吾妻 3-1

本稿では、日本語文書を点字に翻訳する問題をとりあげ、文節区切りのためのルールを整理、分類して知識ベース化し、より精度の高い文節区切りを行なう方法を提案する。この方法は文法情報を含む大規模な辞書の代わりに見出し語のみからなる小規模なテーブルを用いることにより、辞書構築の手間と辞書引きにかかる時間を削減することができる。日本語文書を適切な文節に区切る方法は従来より数多く提案されているが、文節区切りの知識がアルゴリズムから独立しておらず、また辞書を作成するために専門家の知識が必要であることから、精度の改良方法が限られている。ここでは文節区切りルールを知識ベース化してアルゴリズムから独立させ、システムとユーザが協調することによって、日本語点字翻訳のための文節区切りを行なうシステムについて述べる。本手法を情報処理関連のテキストに適用し、有効性を確認した。

キーワード 知識ベース、表層解析、日本語点字翻訳、文節区切り

Japanese Sentence Segmentation Algorithms for Translating Japanese to Braille using Knowledge Base

Emiko SUZUKI† Satoshi ONO‡ Taiki HIRAOKA‡ Hitoshi KANOH‡ Seiichi NISHIHARA‡

†Tokyo Kasei Gakuin Tsukuba Junior College
‡Institute of Information Science and Electronics, University of Tsukuba
Azuma 3-1, Tsukuba 305, Japan
emiko@cs.kasei.ac.jp

Many Japanese sentence segmentation algorithms have been proposed to translate Japanese into English or to query databases. Those methods use a huge dictionary including word representation, readings, and grammar references which require considerable time and work. Since Braille needs only blanks and phonetic information, we do not have to check grammatical combination of words. We propose a new method to segment the Japanese sentence in order to translate Japanese into Braille. Our methods uses a knowledge base which categorizes Japanese sentence segmentation rules. Segmentation rules for translating into Braille are heuristic, ambiguous and complicated. Software is available but the user interface is not very good and volunteers rarely use it. So we provide a user interface for checking the position of ambiguous segmentation. In this way, the users' workload is reduced since it is no longer necessary to check all parts of the sentences. In our method, only a few small tables including words with the segmentation patterns are necessary. Our knowledge base does not need any grammatical information, but utilizes surface information such as Kanji, Hiragana, Katakana, and other character types. The accuracy of segmentaion is 98.0% - a much higher rate than that found in usual methods.

key words Knowledge Base, Surface Analysis, Japanese to Braille Translation, Japanese Sentence Segmentation

1 はじめに

パーソナルコンピュータの普及に伴い、コンピュータを使って障害の一部あるいは、多くの箇所を代行させようという研究が大学をはじめ、企業でも進められている。しかし、その数にしても実用度という点に関しても、日本は欧米に著しく遅れをとっているのが現状である。これは日本語の問題が関わってくる。欧米諸言語のように分かち書きされ、単語と単語の間に空白のある文章と異なり、日本語の文章は字種も豊富であるうえに単語と単語が分かれていない。このため、日本語で書かれた文章を音声合成して読ませようとしても、まず、日本語を文節区切りして単語を認識したうえでなければ正しく読ませることができない。点字翻訳（点訳）においても同様の問題を解決する必要がある。

日本語を点字に翻訳するシステムは過去においていくつか提案されている。日本アイ・ビー・エムの嘉手川らは約77000語の基本単語辞書を用いて分かち書きと漢字かな変換を行なうシステムを開発した[5]。また、筑波技術短期大学の河原は既存の点訳プログラムの誤変換を分析したうえで、ICOTの形態素解析辞書を用い、接続表による解析を行なうシステムについて報告している[2]。

日本語の文節区切りは従来、文節区切り単独の処理としてというよりは機械翻訳や日本語によるデータベース質問等のための形態素解析として佐藤[8]、坂本[3]、長尾[9]、宮崎[7]らによって研究されてきた。これらの方法の問題点としては文節区切りを行なうために用意する辞書を構築するために非常に多くの時間と労力を要したうえ、辞書引きに時間がかかって遅いということである。膨大な辞書を用意して完璧に文節区切りが行なえるというのならば時間がかかっても有効であるが、日英機械翻訳やデータベース検索を行なうのと異なり、点訳の場合、機械的にフィードバックして誤りの可能性を指摘することができない。これは点字の文節区切りのルールが非常に特殊化していることと、日本語の文節が曖昧であるためであるが、必ず人間が見直すという過程がある。この過程が省略できない以上、システムは対話型で処理時間のかからないものが望ましいと考えられる。

ここでは膨大な辞書のかわりにひらがな書きされる自立語テーブル、混ぜ書きされる自立語テーブル、といった従来の辞書に比べると非常に小規模なテーブルのみを用い、文節に区切るか区切らないかをルールで表現し、ルールごとの優先度を考察することによって、従来方法より精度の高い文節区切りを行なう手法を提案する。また、本手法を用いて対話的に文節区切りを行なうシステムを構築したのでそれについても考察する。

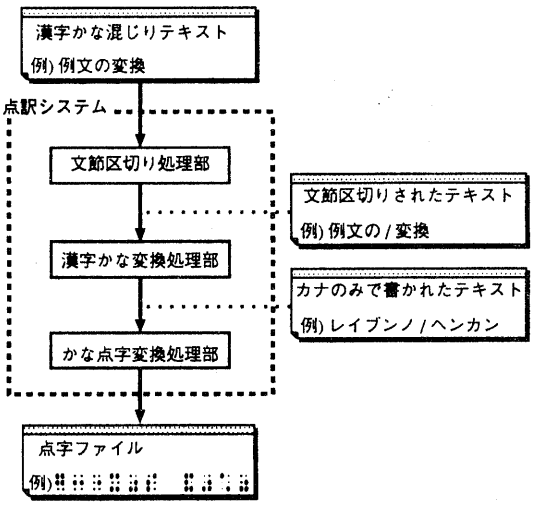

図 1: 点訳の手順

2 従来手法とその問題点

2.1 点訳のための文節区切り

点訳は一般に図1に示したような手順で行なわれる。すなわち、漢字かな混じり文を点字の規則に従って文節に区切ったのち、漢字に読みをつけ、読みを表音文字体系に変換し、最後に点字出力形態に合わせて点字を出力する。

従来この作業は、点訳ボランティアによりすべて手作業で行なわれていた。一字一字点筆を用いて打点するため、修正するにはそのページ全体を打ち直す必要があり、非常に時間がかかっていた。最近、パソコンで動作する点訳プログラムが入手できるようになって点訳の効率は格段に上がった。しかしプログラムでできることには限界があり、点訳プログラムはあまり利用されていない。これにはいくつかの原因が考えられるが、その主なものは、次の2点と考えられる。

- 点字の文節区切りは従来の文節区切りでは対応しきれないほど複雑で曖昧であり、ボランティアは区切られた文章を全部見直す必要がある。
- 日本語の漢字は複数の読みをもつものが多く、読みを一意に決定できないにもかかわらず、点訳プログラムでは一つの読みを与えており、ボランティアは読みについても見直して修正しなければならない。

2.2 従来手法を用いた文節区切りとその問題点

計算機を用いた日本語の文節区切り手法には様々なものがあり、一般には文節区切り単独としてよりは形態素解析、構文解析を伴う音声合成や日英機械翻訳、

日本語文書校正などに利用されている。このような言語処理のための文節区切りでは単語を単語として認定するだけではなく、その後の処理のために形態情報をはじめとする構文情報、意味情報といったものを文節区切りの段階で付与するのが一般的である。

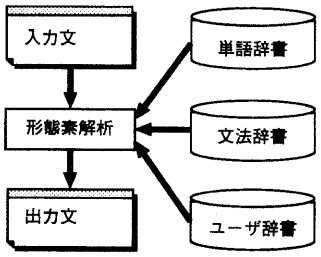

図 2: 従来手法による文節区切り

図 2は代表的な従来手法の構成である。この図に示したように与えられた入力文に対して単語辞書や文法辞書、それにユーザ辞書などを用いて形態素解析が行なわれ、文節と認定された単位で区切られる。

文節区切りには代表的な 2 種類の方法が提案されている [6, 8]。これらは機械翻訳等に広く用いられているが、点訳に適用するには以下のような問題点があると考えられる。

問題点① 形態素解析を行なうため、文法情報が必要となり、辞書が膨大である。

問題点② 精度の改良方法は一般にユーザ辞書への単語登録に限られているにもかかわらず、登録のためのユーザインタフェースが悪い。

問題点③ 文節区切りの誤りをその後の処理で回復することが難しい。

点訳で必要とされるのは、単語ごとの区切り方や単語の読み、といった情報のみであり、従来の文節区切りとは異なった方法が有効であると考えられる。

また点訳の場合、文節区切りを誤るとその段階で点字としての情報の欠落が大きく、その後の処理で回復することは難しい。このため、文節区切りの誤りは文節区切りの処理の段階で訂正されることが望ましい。

2.3 提案する手法の基本方針

前述のような問題点をふまえ、筆者らは従来より行なわれてきた形態素解析に用いられるような大規模な辞書と文法規則を用いず、簡便な表層解析のみを行なうことによって点訳のための日本語の文節区切りを行なうことを試みた。

本手法の基本方針は次の 4 点である。

1. 文節区切りの処理を自動分割と対話処理の 2 段階で行う。
2. 文法情報を含まない単語のみからなる 7 種類のテーブルを用いて表層解析を行なう。
3. 文節区切りの知識を表層解析に基づく知識に書き換え、知識ベース化する (以下、知識ベース A と称する)。
4. 文節区切りが疑わしい箇所をユーザに提示するための知識を知識ベース化する (以下、知識ベース B と称する)。

3 表層解析に基づく文節区切り

ここでは、まず表層解析と文節区切りに必要な知識について説明し、知識の獲得方法とテーブルの構築方法について述べる。次に知識ベース化について述べ、最後にシステムの構成と文節区切り手順について説明する。

3.1 表層解析

3.1.1 表層情報

表層情報を、漢字かな混じりテキストの字面から判別できる字種や以下の節で述べる語句情報と定義する。この情報は後述するテーブル名の項目と一致する。また、与えられた文からこれらの表層情報を抽出することを表層解析と呼ぶ。

3.1.2 各種テーブル

表層解析で用いるテーブルを表 1に示す。

表 1: 表層解析で用いるテーブルの一覧

テーブル名	書式	語数	容量 (kbyte)
ひらがな書き自立語	単語, 区切り方	250	0.3
助詞	単語	25	2.3
漢字 2 字熟語	単語	62,500	312.6
漢字 3 字熟語	単語	24,000	169.3
接頭語	単語	20	0.1
接尾語	単語	60	0.2
混ぜ書き語	単語	11,000	75.8

本手法で用いるテーブルはここに示した 7 つのみで、従来の形態素解析辞書と決定的に異なるのは、文法情報をもっていない点である。ひらがな書きの自立語テーブル以外はすべて単語のみからなる。ひらがな書き自立語のみが、そのひらがなの「前後で区切る」か「その単語の内部で区切らない」か、「前で区切る」か、あるいは「後ろで区切る」かの情報をもっている。

これらのテーブルはEDR日本電子化辞書研究所の「日本語基本単語辞書」から単語を抽出して作成した。ひらがな書き自立語については情報処理関連の文献に出現しない単語を削除し、残った単語について、区切り方の情報を人手により入力した。また、接頭語、接尾語については文献[9]を参照し、実際のテキストから抽出して作成した。ひらがな書き自立語以外のテーブルは単語のみからなるため、語の追加・削除が容易である。また、ひらがな書き自立語についても区切り方を区別するだけですみ、ユーザが容易に修正できる。

3.2 文節区切りのための知識

3.2.1 知識の分類

文節区切りに必要な知識の例を表2に示す。知識の総数は現在のところ39個である。

表2: 文節区切りのための知識の例

分類	知識番号	知識	優先点数
句読点	1-1	句読点の前で区切らない	9
	1-2	句点の後ろで2回区切る	10
	1-3	読点の後ろで区切る	9
自立語	5-1	{前後 or 前}を区切る種類のひらがな書き自立語の前を区切る	7
	5-2	{前後 or 後}を区切る種類のひらがな書き自立語の後ろを区切る	5
	5-3	ひらがな書き自立語の内部は区切らない	4

1. **句点、かっこ、スペースに関する知識**
   点訳のための文節区切りを行なう上で、必ず守らなければならない知識を導入した。これらの知識には高い優先点数を与えた。

2. **字種の変わり目に関する知識**
   字種の変わり目に着目した知識を導入した。様々な字種の組合せについて12個の知識で対応する。

3. **漢字熟語、ひらがな書きの自立語、混ぜ書き語、助詞、接頭語・接尾語に関する知識**
   字種の情報だけでは区切り方の曖昧な箇所について各種テーブルから得られる表層情報を基に区切り方を決定するための知識を導入した。

4. **その他に関する知識**
   上記1〜3を用いても区切り方が曖昧な箇所に対して導入した。

3.2.2 優先点数

各知識に優先点数を設定した。これは、知識が競合した場合にどの知識を選択するかを決定するためのもので、各知識間の優先度の強弱関係をすべて満たすように設定した。各知識間の相対的な強弱関係の例を表3に示す。例えば、「1-2<2-2」は、1-2の知識より2-2の知識の方が優先度が強いことを表している。優先点数の決定手順を以下に示す。

表3: 知識間の優先度の相対的な強弱関係の例

1-2 < 2-2	1-3 < 2-2	2-3 < 2-1	2-3 < 3-2
9-1 < 1-3	3-1 < 6-2	3-1 < 10-5	3-3 < 8-2
3-7 < 10-2	3-9 < 10-3	3-9 < 10-4	2-4 < 8-2
2-4 < 10-3	7-1 < 1-2	7-1 < 2-1	7-1 < 9-2
7-1 < 3-11	7-1 < 7-3	7-1 < 8-3	7-1 < 10-3

STEP1 各知識間の優先度の強弱関係を決定する。

STEP2 強弱関係を満たすように優先点数を設定する。もし、強弱関係を表す不等式に循環が生じた場合、次のようにする。

STEP2-1 出現頻度の高い組合せの強弱関係を優先して点数を決定する。

STEP2-2 出現頻度で一意に決定できない知識に関しては点数を同点とし、対話処理の段階でユーザに区切り方を問い合わせる。

3.3 知識の獲得とテーブルの構築

知識の獲得とテーブルの構築は、通常のエキスパートシステムにおける知識ベースの構築手順[1]に従って以下のように行なった。

STEP1 点訳ボランティアへのインタビューや文献調査から、まず2種類のテーブル(ひらがな書きの自立語、助詞)と16個の知識を構築した。

STEP2 実際の文献に適用して誤った箇所を解析した。

STEP3 誤りが検出された各箇所についての知識を追加・更新・削除した。同時にテーブルの単語の見直し、テーブルの追加を行った。

以上のSTEP2からSTEP3を繰り返すことにより、最終的に7種のテーブルと39個の知識を得た。

3.4 知識ベース化

提案する手法は、各文字間に区切りを入れるかどうかというルールを知識ベース化することで、知識の追加、削除、更新を容易に行なえるようにした。

ここでは文節区切り問題を定式化し、例えば表2の知識1-3は以下のように簡潔に表現することができるようにした。

知識1-3 R_3 = (1-3 , 9 ,
if (後ろの文字 = 読点) ,
then 区切り方 = 区切る)

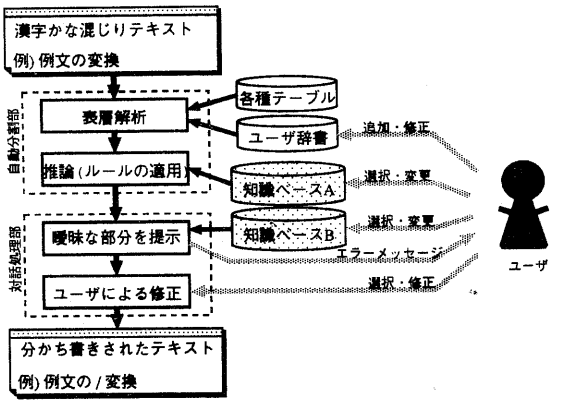

図 3: 本手法に基づく文節区切りシステム構成図

3.5 システムの構成と文節区切り手順

本手法を用いて文節区切りを行なうシステムを開発した。図 3にシステムの構成を示す。本システムは自動分割部と対話処理部の 2 つから構成される。

文節区切りは、次に示すような手順で行なわれる。

STEP1 [自動分割部] 入力文から各種テーブルと字種情報を用いて表層情報を抽出し、表層情報をもとに、知識ベース A の知識を用いて、文節区切り箇所を決定する。

STEP2 [対話処理部] 知識ベース B の知識に基づいて曖昧な区切り箇所をユーザに提示し、見直しを促す。

自動分割部における知識ベース A の適用と処理手順を図 4に示す。

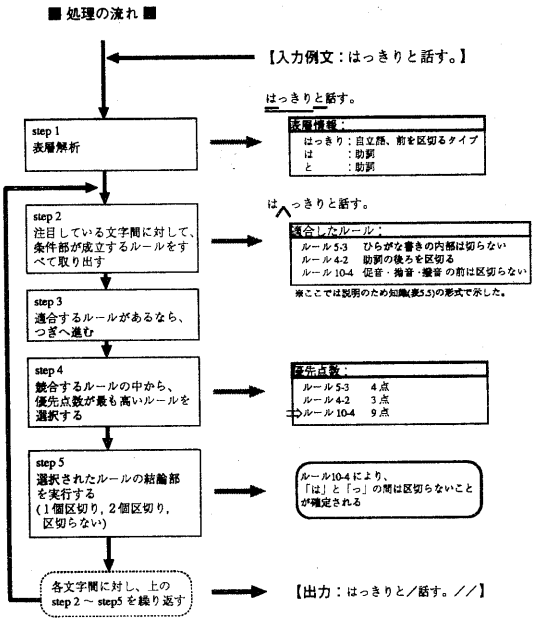

図 4: 自動分割部における文節区切り処理手順

4 実験

4.1 実験方法

テキスト [4] を用いて本システムの性能評価実験を行なった。これは文章数 3463 文、文字数にして 113884 文字の情報処理関連のテキストである。文節区切りの正解率は次式によって計算した。なお、空振りとは区切る必要のない箇所を区切ってしまった間違い、見逃しは区切らなくてはならないのに区切らなかった間違いとする。

$$\text{正解率（空振りなし）} = (1 - \frac{\text{空振りの回数}}{\text{正解の区切り数}}) \times 100$$

$$\text{正解率（見逃しなし）} = (1 - \frac{\text{見逃しの回数}}{\text{正解の区切り数}}) \times 100$$

また、「ユーザインタフェース」のようなカタカナ連続の区切りは、点訳ボランティアの主観に依存するという理由から、正解率の計算には含めない。本実験は IBM Aptiva model H55 上で行なった。

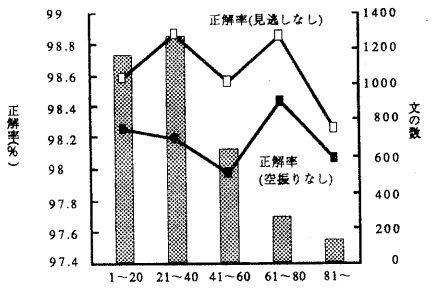

図 5: 文の長さと正解率

4.2 実験結果と考察

4.2.1 文の長さと正解率の関係

図 5に文の長さと正解率の関係を示す。折れ線グラフは正解率、棒グラフは文の数を表す。一般に形態素解析による文節区切りは、文の長さが長くなれば長くなるほど候補となる単語の組合せ数と曖昧さが増加し、正解率が低下するといわれている。これに対して本手法による文節区切りは文の長さにあまり依存せず、高い正解率を保っている。

4.2.2 市販の点訳プログラムとの比較

表 4に市販の点訳プログラム（EXTRA Ver.3.0) と本システムの文節区切りの精度の比較を示す。文節区切り手法は公表されていないが、形態素解析を行なっ

ているものと思われる。前述のテキスト1冊について、空振りなし正解率、見逃しなし正解率ともに本手法のほうが優れており、本手法の有効性を確認できた。ここで、ひらがな書き自立語テーブルと接頭語・接尾語テーブルは実験用テキストに出現する語がすべて含まれるように構築したが、他のテキストに対しても容易に適用できると考える。

4.3 対話処理部の実行例

図6は本手法を用いた文節区切りシステムの実行画面の例である。これは自動分割処理の後、対話処理で修正を行なっているところである。ここで、システムは知識ベースAの知識をもとに区切った箇所を'/'で示し、同時に知識ベースBを用いて区切り方の疑わしい箇所を三角と網かけで示している。ユーザは上記のような記号で示される箇所について、図に示されたようなダイアログボックスにより、どのような知識によって区切られているか(いないか)、疑わしいとされているかを知ることができ、必要に応じて修正を行なう。

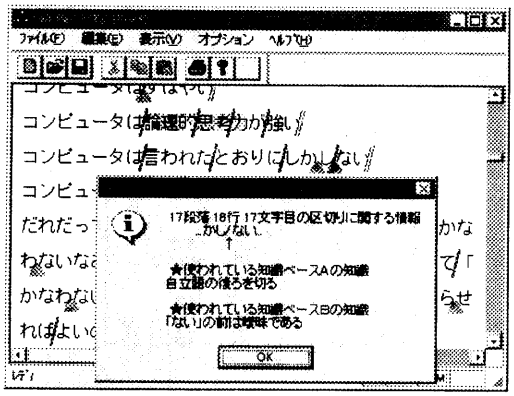

図 6: システムの実行例

本システムは一般のボランティアが使用することを考慮してVisual C++を用いてWindows 95上で動作する対話型システムとした。文節区切りは段落単位で、あるいは全文書を一括に処理することができ、対話の応答時間はユーザの思考を妨げない程度のものである。現在のところ、知識ベースBによってみつけることのできる誤りは誤り全体の約4分の1程度であり、約5%については正確に誤り箇所を指摘することができた。今後、この知識ベースBを充実させることにより、さらにユーザインタフェースの向上がみこまれる。

5 おわりに

今回、様々な文法情報を含む大規模な辞書の代わりに単語のみからなる小規模なテーブルを用いて日本語の文節区切りを行なう手法について提案した。この方法は表層情報に基づく文節区切りのルールを知識ベース化し、知識に優先度をつけることによりルールの適用順位を変化させることができる。さらに、この手法を適用した自動分割部と、曖昧な区切り箇所についてはユーザに問い合わせる対話処理部からなるシステムを構築した。

視覚障害者の職域拡大にはコンピュータの利用技術を身につけることが非常に有効であり、そのためには情報処理関連の専門書の点訳が必須である。にもかかわらず、一般図書と比べて専門書は市販の点訳プログラムで翻訳すると誤りが多く、かえって点訳ボランティアの手を煩わせているのが実状である。このような理由により今回は情報処理関連の専門書の点訳について述べたが、提案する手法はテーブルの変更により、他の分野にも容易に適用できる。

表4: 本システムと他のシステム(EXTRA Ver.3.0)との正解率の比較

	本システム	EXTRA
正解率(空振りなし)	98.2%	95.40%
正解率(見逃しなし)	98.7%	96.03%

謝辞

本研究を進めるにあたって有意義なコメントを頂いた筑波大学理工学研究科の西森雄一氏、水野一徳氏をはじめ、筑波大学非数値処理アルゴリズム研究室の皆様に感謝いたします。

参考文献

[1] Frederick Hayes-Roth, et.al.,"エキスパート・システム", 産業図書,1985.

[2] 河原正治,"日本語自動点訳ソフトウェアの開発について", ヒューマンコミュニケーション研究会 hc94-49,1992.

[3] 坂本義行,"日本語形態素解析の基本設計", 自然言語処理研究会,1983.

[4] 宇都宮公訓,"コンピュータ入門", 共立出版,1990.

[5] 嘉手川繁三, 脇田修躬, "日本語点訳システム-漢字かな混り文から点字まで-", 情報処理学会第27回全国大会, pp. 1237-1238,1973.

[6] 吉村賢治, 日高達, 吉田将, "文節数最少法を用いた日本語文の形態素解析", 情報処理学会論文誌,Vol.24, No.1,1983.

[7] 宮崎正弘,"係り受け解析を用いた複合語の自動分割法", 情報処理学会論文誌, Vol.25, No.6, 1984.

[8] 佐藤大和, 匂坂芳典, 小暮潔, 嵯峨山茂樹, "日本語テキストからの音声合成", 通研実報, Vol.32, No.11,1983.

[9] 長尾真, 辻井潤一, 山上明, 建部周二, "国語辞書の記憶と日本語文の自動分割", 情報処理学会, Vol.19, No.6,1978.

盛光印刷所