自然言語における有繋文字列の抽出

延澤 志保
堤 純也, 孫 大江, 佐野 智久, 佐藤 健吾, 大森 久美子, 中西 正和
慶應義塾大学理工学研究科計算機科学専攻

はじめに

接着語の処理では切り分けの単位として形態素を用いることが一般的だが、形態素は言語学的な単位であり、これが計算機上での処理に適しているかについては十分な議論がされていない。

本稿では、計算言語学上の接着語の意味単位として有繋文字列 (linky strings)[1] を提案し、その特徴を報告する。有繋文字列は日本語のみを対象にしたものではないが、本稿では日本語について考察する。

1 有繋文字列

本稿では、用法や文法などの知識を一切利用せず、統計情報のみによって意味のある文字列の抽出を行なう。本稿の手法で抽出される文字列は統計情報から繋がって出現する可能性が高いと判断されるもので、これを有繋文字列と呼ぶことにする。また、ある文字の並びが同時に並んで出現する可能性が高いと判断される場合、この文字列は有繋性が高いといい、有繋性の高さを数値で表したものを有繋評価値と呼ぶ。

2 有繋評価値の算出

有繋評価値の算出には、コーパスから得た d-bigram 統計情報を用いている。距離 d を考慮する相互情報量 MI は、式 1 のように定義される [2] 。

$$MI(x_i, x_{i+d}, d) = \log_2 \frac{P_w(x_i, x_{i+d}, d)}{P(x_i)P(x_{i+d})} \quad (1)$$

ここで、x_i は事象を、d は事象間の距離を、$P(x)$ は事象 x が起こる確率を、$P_w(x_i, x_{i+d}, d)$ は事象 x_i と事象 x_{i+d} の距離 d での同時出現確率である。

Automatic Extraction of Linky Strings in Natural Languages. Shiho Nobesawa, Junya Tsutsumi, Sun Da Jiang, Tomohisa Sano, Kengo Sato, Kumiko Oomori and Masakazu Nakanishi. Department of Computer Science, Keio University.

MI を用いて、隣合う 2 文字間の有繋評価値を算出する式を以下のように定義する (式 2) [1] 。

$$UK[i] = \Sigma_{d=1}^{d_{max}} \Sigma_{j=i-(d-1)}^{i} \frac{MI(w_j, w_{j+d})}{g(d)} \quad (2)$$

ここで、w_i はある文中の i 番めの文字、d は 2 文字間の距離、d_{max} は文字間距離の最大値、$g(d)$ は距離に対する重み付け関数を示す。文中のある 2 文字の有繋評価値はその前後の文字の d-bigram 統計情報をも加えて算出される (図 1) 。

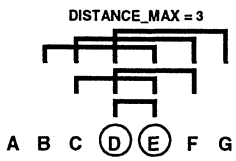

図 1: 有繋評価値に影響する d-bigram データ

3 システム概要

本研究で作成したシステム *LSS*は、d-bigram 統計情報を用いて入力文中の隣り合う 2 文字間の有繋性を調べることで自動的に入力文の切れめを作成する。

文中のある 2 文字間の有繋評価値には、その 2 文字の前後の文字についての情報も影響を与えている。これは、例えば「もし」という 2 文字の並びは「おもしろい」に含まれる場合には切り分けるべきではないが、「とてもしずかだ」のように必ずしもひとかたまりとは限らない。ある 2 文字の間を切り分けるべきかの判断は、その 2 文字の同時出現確率だけでなく前後の情報をも利用する必要がある。有繋評価値の算出においては離れた文字間の関係もとらえることができる d-bigram を用いることで、切り分けにおける正解率の向上をはかる。

ある文中の各文字間の有繋評価値をグラフで表すと、次のようになる (図 2) 。図 2 のように、ある文を構成する各文字間の有繋評価値をグラフで示したものを評価値グラフ (a score graph) と呼ぶことにする。評価値グラフでは x 軸は文中の各文字、 y 軸は各文字間の有繋評価値を示す。

図 2: 評価値グラフ

有繋評価値が高いほどその 2 文字は関係が強い。グラフ中の谷の部分は評価値が低くなるところなので、文字列の切れめを示す。従って、この谷を見つけることで文字列の切れめを推測することが可能である。

以下に評価値グラフによる切れめの作成規則を示す。

A. 山の部分では切り分けない。
B. 文頭は必ず文字列の最初とする。
C. 一文字有繋文字列は前後に切れめを入れる。
D. グラフの谷の部分で切り分ける。
E. 文末文字の前で切り分けるかを決定する。
F. それ以外の場合には切り分けを行なわない。

先に述べたように、*LSS*は評価値グラフの山の部分を有繋文字列と認識して抽出する。基本的に、山とは図 2 の A-B や C-F のように、評価値の谷から次の谷までの範囲に含まれる文字列を指す。

評価値グラフ中の谷の部分は有繋文字列の切れめを示し、*LSS*は、V 字の底の部分を切り分け箇所とする。

一文字から成る有繋文字列は山を形成することができない。そこで、評価値グラフ中の谷の部分で平な辺がみつかった場合、これを一文字有繋文字列と判断することにする。

4 実験

ここでは、有繋文字列切り分けシステム *LSS*を用いた日本語漢字かな混じりべた書き文の切り分けについての実行結果を示す。本研究では 1991 年の朝日新聞の記事 7,502 文を training corpus として用いた。

実験 1　d-bigram の有効性の調査

実験 2　open corpus での妥当性の調査

4.1 d-bigram の有効性の調査

ここでは closed corpus 302 文について d-bigram 情報を用いて切り分けを行なった場合と bigram 情報のみで切り分けを行なった場合とを比較する (表 1) 。

表 1: 出力結果中の有繋文字列の数

	d-bigram $d_{max}=5$	bigram
入力文数	302	302
有繋文字列数	6,145	7,098
1 文あたりの有繋文字列数	20.35	23.50
過分割箇所数	454	689
過分割の割合	7.39%	9.71%

表 1 は抽出される有繋文字列の過分割の割合が非常に低いことを示している。d-bigram の結果では、1 文あたりの平均有繋文字数は 20.35 個であるのに対して 1 文あたりの過分割数は約 1.4 個と少ない。

過分割は主にひらがな文字の並びの中で起きることが多い。全過分割数に対するひらがな文字間の過分割数の割合は、 d-bigram を用いた場合で 73.3% にのぼる。次に頻度の高い漢字文字間の過分割数が全体の 13.0% であることに較べると、ひらがな文字間の過分割の割合が圧倒的に高い。

bigram を用いると d-bigram を用いた場合に較べて文が細かく分かれる傾向がある。同一の入力文に対する出力結果[1]を見ると、 bigram を使った方が抽出された有繋文字列の数も多く、 1 文あたりの有繋文字列数も d-bigram の場合の 115.5% になる。bigram では長い有繋文字列が切れやすいため、このことが 1 文中の有繋文字列数の差に繋がっていると考えられる。

4.2 open corpus での妥当性の調査

同じ新聞から得た closed corpus と open corpus との切り分け結果の比較を行なう。以下に closed corpus と open corpus の切り分け出力結果[1]を示す (表 2) 。

表 2: 出力結果中の有繋文字列の数

	closed corpus	open corpus
入力文数	302	300
有繋文字列数	6,145	7,404
1 文あたりの有繋文字列数	20.35	24.68
過分割箇所数	454	671
過分割の割合	7.39%	9.06%

[1]closed corpus での結果は、先の d-bigram の有効性の調査での d-bigram を用いた結果と同じ。

open corpus では名詞および固有名詞での過分割が closed corpus に対して非常に多い。字種毎では漢字文字の前後での過分割の個数が多くなっている。open corpus による実験で過分割と判断された文字間の切れめの中で漢字の前後で起きたもののうち 94.17% がその 1 回だけ過分割されたものである。漢字文字は表意文字であり 1 文字で具体的な意味を持つ多いため、いろいろな文字と繋がって出現する可能性が存在する。そのため出現回数の少ない並びやいろいろな文字と繋がりを持つ漢字文字が多く、open corpus では未知の繋がりが増えるため漢字文字の前後で過分割が起こる割合が高くなっている。

漢字の特性による過分割の増加を除けば、open corpus での切り分け結果はおおむね closed corpus のときと同様である。漢字文字の前後での過分割の問題は training corpus の選び方である程度解決できるので、open corpus においても十分な成果を得ることができると言ってよい。

5 有繋文字列の性質

5.1 統計情報のみを用いた切り分け

test corpus に対して *LSS*による切り分け処理を施した結果、過分割の数は抽出された有繋文字列の数に対して 7.39% でしかない (表 1 d-bigram) 。*LSS*は統計情報のみを用いて切り分けを行なっており、文字の意味や用法などに関する知識は全く利用していない。このことから、統計情報のみによる方法でもべた書き文の切り分けは十分に可能であると言える。

5.2 助詞の扱い

抽出された有繋文字列を見ると、助詞がその前後の形態素に付着しているものが多く見られる。例えば、「海外」という文字列を含む有繋文字列は以下のようである (表 3) 。

表 3: 「海外」を含む文字列と助詞の関係

海外	海外に	海外派遣	海外派兵	海外旅行の
	海外へ	海外派遣は		
		海外派遣へ		
		海外派遣を		

「海外」という形態素の意味を考えると、後に「に」と「へ」という行き先を示す助詞が付きやすいというのは、納得できる。同じ「海外」を含む有繋文字列でも、「海外派遣」という文字列に対しては、「は」「へ」「を」の有繋性が高い。このように、助詞を含む有繋文字列は核となっている形態素の使われ方がある程度特定されることを表す。

抽出された有繋文字列中の中で「を + 動詞」の形をしたものをいくつかあげてみる (表 4) 。

表 4: 助詞「を」を含む文字列の一部

をえなかった	を逸する	を歓迎した	を指摘した
をする	を引き起こ	を含め	を示す
をつく	を越え	を繰り返	を持つ
をめざす	を確認し	を呼ぶ	を受け

「持つ」は基本的に動作の対象となる目的語、ここでいう「～を」に当たる部分を必要とする。従って、これに「を」がついた「を持つ」で 1 有繋文字列となることは自然である。

5.3 活用語の切り分け

抽出された有繋文字列の中で過分割されたものの特徴のひとつに、動詞の語尾での過分割が挙げられる。

ここでは、特に過分割されやすい上一段活用と下一段活用について考えてみる。例えば「感じる」を例にとると、この動詞が出現した場合にはどの活用形でも「感じ」までの 2 文字が必ず並んで現れるため、「感」という文字と「じ」という文字との間の有繋性は高くなる。それに対し、「感じる」の「じ」と「る」の繋がりは、必ず起こるという保証はない。従って、有繋性に応じて切り分けを行なった場合、「感」と「じ」の間よりも「じ」と「る」の間の方が切れやすくなる。これは、統計的に見た場合には、上一段動詞は文法的に定義されるように「感」を語幹、「じる」を語尾とするのではなく、「感じ」までを語幹、「る」などその後に続く部分を語尾と考える方が合っているということができる (表 5) 。

表 5: 有繋性による上一段活用の語幹と語尾の分割

語幹	未然	連用	終止	連体	仮定	命令
感じ	—	—	る	る	れ	ろ

下一段活用は語尾の第一字めが「え」段であるということ以外は上一段活用と同じなので、上一段動詞と全く同じことが下一段動詞についても言える。語尾の

第一字めを語幹の一部とみなすことは、意味的には全く問題がない。従って、有繋文字列の抽出という観点から見た場合、上一段動詞および下一段動詞は表 5 のように扱うことが妥当である。

実験結果中では国語辞典を基準として評価しているため、語尾が切り分けられる例はすべて過分割として扱っているが、ここで述べたように上一段動詞と下一段動詞の切り分けの見方を変えることで過分割とされる事象は減少する。

5.4 「決まった言い回し」の抽出

*LSS*では決まった言い回し (fixed locution) を抽出することが可能である。抽出結果では、「手を貸す」「手を打つ」「手を差し (出す、のべるなど)」「手を抜く」「手を切る」「手をつけ (る)」など、実際の行動の記述というよりは言い回し的な文字列が多く抽出されている。言い回しとして頻繁に使われるものは頻度が高くなり、抽出されやすい。

このように、決まった言い回しを抽出することができることは、自然言語処理では非常に有用である。機械翻訳において、決まった言い回しは単純な形態素の置換で翻訳することができない (図 3) 。

日本語文	彼は私に手を貸した
単純置換による英訳例	he lent a hand to me
正しい英訳例	he helped me

図 3: 決まった言い回しの翻訳

このとき、「手を貸す」という日本語を 1 つの有繋文字列として扱うと、この有繋文字列に対して 1 つの対訳単語 (あるいは熟語) を割り当てることができるようになり、翻訳の精度が増すものと考えられる。この場合、「手」と「貸す」もそれぞれ有繋文字列として登録されている可能性が高いので、実際に「手 ("hands") 」を「貸す ("lend") 」する、という候補も同時に生成可能である。

また、形態素解析においても、「てをうつ」というひらがな文字列から「手」と「打つ」、「撃つ」などの関係などを調べることなしに「手を打つ」という有繋文字列を直接導出することが可能になる。

6 結論

接着語の処理単位としては形態素が用いられることが一般的だが、形態素は言語学の立場から定義されているものであり、計算機での処理にふさわしい単位であるかについては十分な議論がなされていない。

本研究では計算言語学的な処理単位として有繋文字列の概念を導入した。文法などの知識情報を一切必要としないためその抽出は容易であり、しかも十分に意味内容を保持している。さらに、形態素レベル (「手」「貸す」など) から熟語レベル (「手を貸す」など) まで同時に取得可能であることも、意味情報の利用の面で期待が持てる。

具体的には機械翻訳における辞書作成の対訳変換の単位としての利用や、文生成における「自然な文の作成 (最良文問題)」のための語句単位、あるいは文書等のキーワード (キー文字列) の抽出など、有繋文字列の概念を用いることで効果を期待できる処理は多く存在する。

今後、統計情報を有効に活用できる有繋文字列の概念の利用によってさまざまな処理の効率および精度の向上が期待できる。

参考文献

[1] 延澤 志保. 自然言語における有繋文字列の抽出. 慶應義塾大学修士論文, 1996.

[2] 堤 純也, 新田 朋晃, 小野 孝太郎, 延澤 志保. 統計情報を用いた多言語間機械翻訳システム. 人工知能学会研究会, pages 7–12, 1993.

[3] Tsutsumi, J., Nitta, T., Ono, K., Nobesawa, S. and Nakanishi, M. Multi-lingual Machine Translation Based on Statistical Information. *Qualico*, pages 147–152, 1994.

[4] Sun, D. J., Tsutsumi, J., Nitta, T., Ono, K., Nobesawa, S. and Nakanishi, M. An intelligent Chinese input system using statistical information between words. *Qualico*, pages 102–107, 1994.

[5] Nobesawa, S., Tsutsumi, J., Nitta, T., Ono, K., Sun, D. J. and Nakanishi, M. Segmeting a Japanese Sentence into Morphemes Using Statistical Information between Words. *Coling-94*, pages 227–233, 1994.