ベイズ教師なし文境界認識

内海 慶 [1]　持橋 大地 [2]
[1]SB Intuitions 株式会社　[2] 統計数理研究所/国立国語研究所
kei.uchiumi@sbintuitions.co.jp　daichi@ism.ac.jp

概要

文を単位とする言語現象を捉えるためには**文分割** (文境界認識) が必要となるが, 実際に我々が扱うことが多い崩れたテキストには無数の文末表現があり, 通常の教師あり学習で正しく分割することは難しい. 本研究では, 文分割をテキストの文字毎に存在する, **二値潜在変数の推定問題**ととらえる. セミマルコフモデルの枠組みで動的計画法と MCMC 法を組み合わせることにより, 単純な文字nグラム言語モデルを用いるだけで, 最新の大規模言語モデルによるヒューリスティックな文分割を超える, **最高精度の文分割を教師なし学習で行える**ことを示す.

1　はじめに

文分割, または文境界認識は翻訳, 構文解析, 含意関係認識などの**基礎として必要**な, 自然言語処理における最初の重要なステップである. 京大コーパスや Penn Treebank といった既存の整備されたコーパスは既に文に分割されていることがほとんどであるが, 図 5 に示したように, SNS などでみられる実テキストは必ずしも文に分けられておらず, **文分割**が最初の処理として必要になることが多い.

日本語では句点「。」が文末を示しているとされるものの, 図 5 のような実際のテキストには句点がないことも多い. また文末表現に関しても記号や絵文字, アスキーアートなどを含めて多様であり, これをルールや人手による教師データから認識することには, カバレージの意味から限界がある.

それでは, 文境界を**教師なし学習**することはできるのだろうか. 文分割の研究の多くは教師あり学習を基にしており, これには人手で大量の「正しい文」のデータを用意する必要がある. しかし, テキストの途中にある文境界は明示されないとしても, ツイートやパラグラフの終わりは必ず文境界であり, **文境界のヒントは生コーパスにおいても, かなり与えられている**と言ってよい. また, 改行が文境界であるこ

ひゃっはああぁぁぁぁぁぁぁあ
きんっきんに冷えてやがるぜ

雪音さん、おやすみなさーい… 🌛💤
配信楽しかったです〜新クリーチャーっ‼

うーん('ω')
ものをつくるひとは
命をかけてそれをつくってる('ω')
気持ちを込めてつくったら
パワーが込められている。
んだなぁ('ω')

疲れた。楽しかった。忙しくて疲れるなんて久々だよ。\
家帰るの面倒くさいよ。

図 1: X での実テキストの例. 様々なタイプの文末があり, 改行の場所以外にも文末が存在する. \で行の継続を示した.

とはかなり多いため, これも重要な**補助情報**である. 特に SNS のようなテキストは X (Twitter) のように短く区切られていることが多いため, こうした文境界のヒントが豊富である. これを上手に活用すれば, 必ずしも人手で教師データを作成しなくとも, **教師なし学習すなわち言語モデルによって**, われわれが無意識に行っているような**文分割を自動的に行う**ことが可能ではないかと考えられる.

そこで本研究では, 文分割の問題を文字ごとの潜在的な二値変数の推定問題として統計的に定式化する. **文字nグラムモデルを統計的に正しく用いるだけで**, SNS のような崩れたテキストに対しても, 既存の擬似教師あり学習や巨大な LLM を用いた文分割を超える性能を達成可能であることを実験的に示す.

2　文境界認識とそのモデル

教師あり学習　文分割は基礎的なタスクであり, 文字または単語に, そこが文境界であれば 1, そうでなければ 0 を付与する二値分類問題とみなすことができるため, 多くの方法は教師あり学習で学習されてきた. 2021 年に発表された Ersatz [1] は Transformer をベースにした識別学習を用いて多くの言語に対応し, ニュースのような標準的なテキストに対しては

99%以上の高い性能を見せることが示されている.

一方で, 識別学習は与えた正解データに依存するため, カバレージを広くできないという問題があり, ルールベースの手法も依然として有用である. 機械翻訳ツールキットである Moses に含まれる文分割器[1)]は高い精度を持っており, 最近発表された PySBD [2] はカバーする文分割のパターンを網羅するようにルールを設計することで, 多くの言語と広い範囲のコーパスに対して高い性能を持つことが報告されている.

教師なし学習 ただし, SNS のような非標準的なテキストに対しては, 上記の二者ともに万能ではない. こうした場合には教師なし学習が有効であり, 実際に NLTK で使われている Punkt [3] はコーパスから教師なし学習で学習されるモデルである. また 2023 年の *Where's the Point* (WtP) [4] は, 改行情報を用いた自己教師あり学習によって文境界を推定する. しかし, 前者はピリオドが文末となる欧米語に特化した発見的な検定量を用いる方法で, 後者は改行位置のみが文境界になりうるという強い仮定を置いており, いずれも日本語のように, **行内も含めて様々な文字が文末になりうる場合**には対応していない.

大規模言語モデル (LLM) を用いる方法として, 2024 年に提案された *Segment Any Text* (SAT) [5] は, LLM に「次のテキストを文に分割せよ」と指示を与えるだけで, 非標準的なテキストについても従来の [4] を超える最高性能の文分割性能を報告している. しかし, これには Llama-3-8B のような巨大な LLM が必要である上, LLM が必ずしも入力テキストを正確に出力するとは限らないという問題もある.

SNS テキスト 日本語で改行が必ずしも文境界とならない SNS のようなテキストに対しては, 林部ら [6] が正解データを内製して教師あり学習を行い, 対応する日本語文境界判定器 Bunkai をリリースしている[2)]. ただし, これには**正解データが必要**なため, 新しい顔文字や文末表現[3)], ノイジーな他言語の場合などに対応することはできないという問題がある.

3 教師なし文境界認識

統計的にみると, 文境界認識とは, 1 章で述べたようにテキストの各文字あるいは単語に, そこが文境界である (=新しい文が始まる) かどうかを表す二値の潜在変数 b が存在し, それを推定するタスクであると考えることができる[4)]. 以下では SNS のような非標準的なテキストへの適用を考えて文字ベースで考えることにし, 言語モデルとしては, 次節で述べる学習時に容易に更新できる文字 n グラムモデルを用いる. 文字列 $s = c_1 c_2 \cdots c_T$ からなる T 文字のテキストの確率は, 文境界を表す潜在変数 $\mathbf{b} = (b_1, \cdots, b_T)$ を周辺化することで,

$$p(s) = \sum_{\mathbf{b}} p(s, \mathbf{b}) = \sum_{\mathbf{b}} p(s|\mathbf{b})\, p(\mathbf{b}) \tag{1}$$

と表すことができる. $\sum_{\mathbf{b}}$ は, 2^{T-1} 個の可能なすべての文分割に関する和である. 式 (1) における $\mathbf{b}$ の可能性は 2^{T-1} 個と指数的に大きいため, 本研究では隠れセミマルコフモデル [9] の考え方を用いて, 動的計画法によって $\mathbf{b}$ を効率的に MCMC 法によってサンプリングし, 推定していく.

3.1 動的計画法による文分割

いま, s の時刻 t までの部分文字列 $s_1^t = c_1 c_2 \cdots c_t$ において, 最後の ℓ $(1 \le \ell \le t)$ 文字が文になっている周辺確率を前向き変数 $\alpha(s_{t-(\ell-1)}^t)$ とおくと, これは定義から, 次のように展開することができる.

$$\begin{aligned}\alpha(s_{t-(\ell-1)}^t) &\equiv p(s_1^t, b_t = 1, b_{t-1} = \cdots = b_{t-(\ell-1)} = 0) \\ &= p(s_{t-(\ell-1)}^t, s_1^{t-\ell}, b_t = 1, \sim) \\ &= p(s_{t-(\ell-1)}^t | b_t = 1, \sim) p(b_t = 1, \sim) p(s_1^{t-\ell})\end{aligned} \tag{2}$$

ここで $\sim$ は $b_{t-1} = \cdots = b_{t-(\ell-1)} = 0$ を表す. α の定義より,

$$p(s_1^{t-\ell}) = \sum_{j=1}^{t-\ell} \alpha(s_{t-\ell-(j-1)}^{t-\ell}) \tag{3}$$

であることに注意すると, 式 (2) は $q = p(b_t = 1)$ とおいて,

$$\begin{aligned}&\alpha(s_{t-(\ell-1)}^t) = \\ &p(s_{t-(\ell-1)}^t | b_t = 1, \sim)\, q(1-q)^{\ell-1} \sum_{j=1}^{t-\ell} \alpha(s_{t-\ell-(j-1)}^{t-\ell})\end{aligned} \tag{4}$$

と再帰的に計算することができる. α が時刻 T まで求められれば, 前向きフィルタリング–後向きサンプリング (FFBS) 法 [10] を用いて, 末尾から文分割 $\mathbf{b}$ をサンプリングする MCMC 法を行うことができる. 学習が終了すれば, 最適な文分割は Viterbi アルゴリズムを用いて求めることができる.

1) https://pypi.org/project/sentence-splitter/
2) https://pypi.org/project/bunkai/
3) 日本語では, 役割語 [7] の一種として「ナリよ」「ラジね」など無数の文末表現が発明され, 日々更新されている.
4) この問題は教師なし形態素解析 [8, 9] と似ているが, 1) 単語 (この場合は文) 間の遷移確率を考慮する必要がない (二重分節ではない), 2) 長さに実質的な上限のある単語と異なり, すべての可能な文分割を考慮する必要があるため, 探索空間が非常に大きい, という大きな違いがある.

3.2 改行と事前確率

式 (4) において, ある文字が文境界となる**事前確率** $q = p(b_t = 1)$ は, 最も簡単には, ベータ事前分布 $q \sim \mathrm{Be}(\alpha, \beta)$ から生成されたと考えることができる. コーパス全体で $b_t = 1$ となった回数を M, 0 となった回数を N とおけば, q の事後期待値は $\mathbf{b}$ の集合 $\{\mathbf{b}\}$ から,

$$\mathbb{E}[q \mid \{\mathbf{b}\}] = \frac{\alpha + M}{\alpha + \beta + N + M} \tag{5}$$

と学習中に動的に推定できる. ただし, コーパス中で改行があった場所は, 必ずではないが, 文末である可能性が高いと考えられる. 本研究ではこの情報を, b_t **の事前確率として**モデルに組み込む. すなわち, $q \sim \mathrm{Be}(\alpha, \beta)$ にかわり, 使用する q を

$$\begin{cases} q_1 \sim \mathrm{Be}(\alpha_1, \beta_1) & \text{if } c_t \text{ の直後が改行} \quad (6) \\ q_0 \sim \mathrm{Be}(\alpha_0, \beta_0) & \text{それ以外} \quad (7) \end{cases}$$

に分けて考える. 学習中に文分割を行った後, 「改行位置の b_t の中で」1 となった回数を M_1, 0 となった回数を N_1 とすれば, q_1 の事後分布は

$$q_1 \mid \{\mathbf{b}\} \sim \mathrm{Be}(\alpha + M_1, \beta + N_1) \tag{8}$$

となる. q_0 の場合も同様である.

本研究ではさらに, 日本語の場合は句点「。」の場所も文末である可能性が非常に高い[5]という情報を含めるため, コーパス上で句点の位置についても同様にベータ事前分布 $q_2 \sim \mathrm{Be}(\alpha_2, \beta_2)$ を考え, 事後分布を用いた. このように q_0, q_1, q_2 と **3 通りの事前確率**を導入して推定することで, ベイズ統計の枠組みで, **改行や句読点の情報を効果的にモデルに組み入れる**ことが可能になる.

3.3 学習アルゴリズムと高速化

提案法の MCMC 法 (Gibbs サンプリング) による学習アルゴリズムを, 図 2 に示した. 式 (4) の $p(s^t_{t-(\ell-1)} \mid b_t = 1, \sim)$ は, s の長さ ℓ の部分文字列 $c_{t-(\ell-1)} \cdots c_t$ が文をなす確率であり, これは文頭と文末を表す特殊文字 ^, $ を用いて

$$p(s^t_{t-(\ell-1)}) = \prod_{j=1}^{\ell} p(c_{t-\ell+j} \mid h_{t-\ell+j-1}) \cdot p(\$ \mid h_t) \tag{9}$$

と計算される. ただし, h_t は時刻 t までの n グラム文脈であり, 必要に応じて ^ を埋めるものとする.

5) 日本語では「です。。」のように句点が複数続く場合や, 「モーニング娘。」, 「ゲスの極み乙女。」など固有名詞に句点を含む場合, 顔文字の一部である場合などがあり, 句点が必ず文の終わりになるわけではない.

1: 言語モデルを初期化する
2: **for** $j = 1 \cdots J$ **do** // 収束するまで繰り返し
3: **for** n = randperm $(1 \cdots N)$ **do**
4: **if** $j > 1$ **then**
5: 現在の文分割でのテキスト s_n の n グラムカウントをモデルから削除
6: **end if**
7: 式 (4) を用いた FFBS 法で, テキスト s_n の文分割 $\mathbf{b}_n$ をサンプリング
8: 新しい文分割でのテキスト s_n の n グラムカウントをモデルに追加
9: **end for**
10: 分割の事前確率 q を式 (8) の形でサンプリング
11: 言語モデルのハイパーパラメータを更新
12: **end for**

図 2: ベイズ教師なし文分割の学習アルゴリズム.

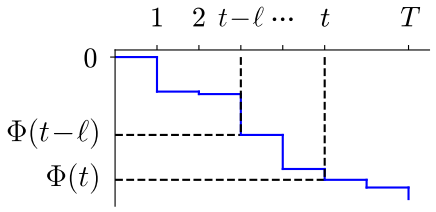

図 3: 高速化のための文の n グラム累積対数確率.

式 (9) の確率は式 (4) の動的計画法の中で, s のあらゆる部分文字列について計算する必要があるため, そのままでは計算量が非常に大きい. しかし, 式 (9) の右辺第 1 項は, 図 3 のように事前に一度だけ対数の累積確率 $\Phi(t) = \sum_{j=1}^{t} \log p(c_j \mid h_{j-1})$ を計算しておけば, $\Phi(t)$ と $\Phi(t-\ell)$ の差を使って $O(1)$ で求めることができる. この結果, $O(T^3)$ だった動的計画法は $O(T^2)$ で計算することができ[6], 大幅に高速化される.

4 実験

SNS の一つである X (以下 Twitter) のツイートおよび, BCCWJ [11] の標準的な日本語文を使って実験を行った. 提案法を, 以下では USBD と表記する.

4.1 データ

Twitter のテキストは, API から取得したランダムな日本語ツイート約 20 万ツイートである. 提案手法は教師なし学習のため, 学習データは任意に増やせることに注意されたい. ロジスティック回帰を用いて, 広告文や機械的に作られた情報など, 文とはみなせないものを除外した. BCCWJ からは Core テキストのうちランダムな 20 万文を学習に用い, さらに 1 万文をランダムに 10 文ずつ連結したテキストを評

6) 同じ工夫は教師なし形態素解析でも適用できるが, 形態素解析では単語の長さは文の長さに比べて非常に短いため, あまり問題になっていなかった.

価データとした. Twitter の評価データは, ランダムな 894 ツイートをクラウドソーシングにより文分割したものである. 5 人の日本人アノテータによる文分割の一致率 (Fleiss の κ 値) は 0.896 であり, 非常に高い. 正解としては文分割の一致率が 80%以上, 100% の二通りに分けて性能を評価した.

4.2 実験設定

言語モデルとしては, 動的な再学習の容易なベイズ n グラム言語モデル [12] を文字ベースで用いた.[7] 本研究では $n=5$ としている. 3.3 節のベータ分布のハイパーパラメータは $\alpha_0=\beta_0=1$, $\alpha_1=\alpha_2=9$, $\beta_1=\beta_2=1$ としたが, 予備実験では性能はハイパラーパラメータにはあまり依存しなかった. 学習では q は式 (5) の期待値を用いて最初の分割を行い, 以下は繰り返し毎に q_0, q_1, q_2 をベータ事後分布からサンプリングしている. 実験は次の二種類で行った.

教師なし学習 Twitter のテキストだけを用いて, 文分割を学習する.

半教師あり学習 BCCWJ のテキストを最初に言語モデルに読み込んでから, Twitter テキストの文分割を行う. 言語モデルは後者の部分のみ更新される.

4.3 実験結果

表 1 に, [4] の WtP および SOTA である, LLM を用いた [5] の SAT との比較を示した. 表 1,2 は一致率 80%の文分割位置を正解としたものであり, 100%の位置についての結果は付録の表 3 に示している. 提案法は WtP のような文末のヒューリスティックや SAT のような教師データを一切使っておらず, 文字 n グラムの簡単なモデルにもかかわらず, 非標準的なテキストの文分割において最高精度を達成している. この精度は, 教師あり学習に基づく Bunkai によるものよりも高い. 図 4 に, Twitter のテストデータを実際に文分割した結果を示した.

BCCWJ の標準的なテキストに対しても提案法は高い文分割性能を達成しており, セミマルコフモデルによる統計的な定式化 (および高速化) と, MCMC 法によるベイズ学習が効果的に働くことがわかる.

4.4 エラー分析

テストデータで文分割を誤った場所を確認したところ, 幾つかの傾向がみられた. 提案法は識別学習で

7) HPYLM の近似である Kneser–Ney 言語モデル [13] を用いても, データ構造を工夫することでモデルを動的に更新することができる [14, 3.4.4 節].

表 1: Twitter テキストの文分割実験の結果 (%).

モデル	説明	精度	再現率	F_1
Bunkai [6]	教師あり学習	78.6	83.3	81.0
WtP [4]	自己教師あり学習	81.3	74.4	77.7
SAT [5]	大規模言語モデル	82.7	**88.8**	85.6
USBD	ベイズ教師なし学習	**85.6**	87.7	**86.6**
	ベイズ半教師あり学習	83.3	93.1	**87.9**

表 2: BCCWJ の文分割実験の結果 (%).

モデル	説明	精度	再現率	F_1
PySBD [2]	ルールベース	66.1	54.3	59.6
Ersatz [1]	Transformer	63.2	44.0	51.9
WtP [4]	自己教師あり学習	84.5	71.6	77.5
SAT [5]	大規模言語モデル	80.0	74.6	77.2
USBD	ベイズ半教師あり学習	**94.7**	**88.7**	**91.6**

はないため, 通常の文の行末の句点や, 箇条書きの行頭を必ず分割できるとは限らない. よって正確には, NPYCRF [15] のような文字特徴量を用いた識別学習との融合が望ましい. いっぽうで, 文末の顔文字や「!」のような記号がアノテータによって文末とされていない場合があり, 評価に用いた「正解」が 100% 正しいとは言えないことに注意が必要である. 文字 n グラムモデルは表層のみを扱っているため, 無数にある絵文字などに効果的に対応するためには, 言語モデルのニューラル化が望ましいと考えられる.

5 おわりに

本研究では, 自然言語処理の最も基礎的なタスクの一つである文分割について, セミマルコフモデルの枠組みで文字 n グラム言語モデルを適切に用いた教師なし学習 (および半教師あり学習) を行うことで, 従来の教師あり学習および LLM による文分割を超える, 最高精度を達成できることを示した.

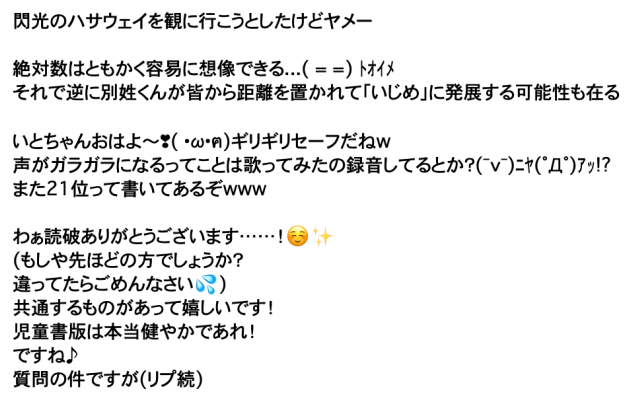

図 4: Twitter テキストの半教師あり文分割の結果. 各行が 1 文に対応している. 元ツイートを付録の図 5 に示した. 顔文字部分などで一部にまだ誤りもある.

参考文献

[1] Rachel Wicks and Matt Post. A unified approach to sentence segmentation of punctuated text in many languages. In *ACL 2021*, pp. 3995–4007, 2021.

[2] Nipun Sadvilkar and Mark Neumann. PySBD: Pragmatic Sentence Boundary Disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pp. 110–114, 2020.

[3] Tibor Kiss and Jan Strunk. Unsupervised Multilingual Sentence Boundary Detection. *Computational Linguistics*, Vol. 32, No. 4, pp. 485–525, 2006.

[4] Benjamin Minixhofer, Jonas Pfeiffer, and Ivan Vulić. Where's the Point? Self-Supervised Multilingual Punctuation-Agnostic Sentence Segmentation. In *ACL 2023*, pp. 7215–7235, 2023.

[5] Markus Frohmann, Igor Sterner, Ivan Vulić, Benjamin Minixhofer, and Markus Schedl. Segment Any Text: A Universal Approach for Robust, Efficient and Adaptable Sentence Segmentation. In *EMNLP 2024*, pp. 11908–11941, 2024.

[6] Yuta Hayashibe and Kensuke Mitsuzawa. Sentence Boundary Detection on Line Breaks in Japanese. In *Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pp. 71–75, 2020.

[7] 金水敏. ヴァーチャル日本語 役割語の謎. 岩波現代文庫 学術 466. 岩波書店, 2023.

[8] Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In *Proceedings of ACL-IJCNLP 2009*, pp. 100–108, 2009.

[9] Kei Uchiumi, Hiroshi Tsukahara, and Daichi Mochihashi. Inducing Word and Part-of-speech with Pitman-Yor Hidden Semi-Markov Models. In *ACL-IJCNLP 2015*, pp. 1774–1782, 2015.

[10] Steven L. Scott. Bayesian Methods for Hidden Markov Models. *Journal of the American Statistical Association*, Vol. 97, No. 457, pp. 337–351, 2002.

[11] Kikuo Maekawa. Kotonoha and BCCWJ: Development of a Balanced Corpus of Contemporary Written Japanese. In *Corpora and Language Research: Proceedings of the First International Conference on Korean Language, Literature, and Culture*, pp. 158–177, 2007.

[12] Yee Whye Teh. A Hierarchical Bayesian Language Model based on Pitman-Yor Processes. In *Proceedings of ACL/COLING 2006*, pp. 985–992, 2006.

[13] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of ICASSP*, Vol. 1, pp. 181–184, 1995.

[14] 持橋大地. 統計的テキストモデル. シリーズ 確率と情報の科学. 岩波書店, 2025 発売予定.

[15] Ryo Fujii, Ryo Domoto, and Daichi Mochihashi. Nonparametric Bayesian Semi-supervised Word Segmentation. *Transactions of ACL*, Vol. 5, pp. 179–189, 2017.

A　図 4 の元ツイート

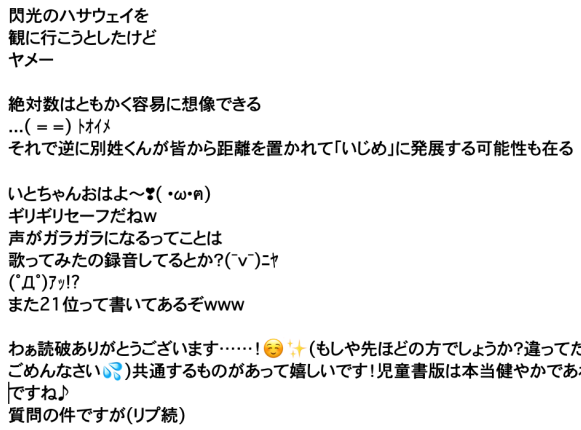

図 5: 図 4 に示した文分割前の元ツイート. 空行でツイートの境界を示している.

B　実験結果の補足

表 3: Twitter テキストの文分割実験の結果 (%). 下表では, 一致率が 100%の箇所を正解とした場合を示した.

モデル	説明	精度	再現率	F_1
Bunkai [6]	教師あり学習	75.6	87.1	80.0
WtP [4]	自己教師あり学習	78.4	83.8	77.6
SAT [5]	大規模言語モデル	77.3	**96.5**	82.2
USBD	ベイズ教師なし学習	**80.9**	90.1	**84.3**
	ベイズ半教師あり学習	78.3	94.9	**84.7**