

ベイズ教師なし文境界認識

持橋大地

統計数理研究所 / 国立国語研究所

daichi@ism.ac.jp

ベイズ統計学シンポジウム
2026-1-23(金)

※ 本研究は、内海慶氏 (SB Intuitions) との共同研究です。

はじめに

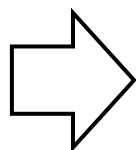
- 自然言語処理＝大規模言語モデル(LLM)に丸投げ、
で終わりか??



- LLMが必ず期待の返答をする保証はない
(LLM自体は、次の単語を返す確率モデル)
- 言語の科学としては、単語や文を取り出して、
そこで何が行われているのかを分析することが
依然として重要

文分割＝文境界認識

今日は寒いね。もう
朝食は食べた？
政府、自治体に輸送
ゼロへの努力を要請



今日は寒いね。
もう朝食は食べた？
政府、自治体に輸送ゼロへの¥
努力を要請

- 文分割：テキストを文に分解するタスク
 - 文以外は含まれていないと仮定 (⇔ 宇田川+(2023))
- 論理関係など、文単位の現象のための基礎技術
- 標準的な文に対しては、高精度なツールが存在
 - Punkt (2006), PySBD (2020), Ersatz (2021)
 - 文末がピリオドで終わるなど、欧米語の強い仮定

文分割＝文境界認識 (2)

ひゃっはあああああああ
きんっきんに冷えてやがるぜ

雪音さん、おやすみなさい… 🌙 zzz

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。
んだなあ('ω')

- 実際のテキストでは、さまざまな文末が存在
 - 「ナリよ」「ラジね」「ねええ💧」
- 文末が句点「。」とは限らず、「。」が文末とは限らない
 - 「モーニング娘。」 「いいひと。」 「(°∀。)

文分割の教師なし学習

ひゃっはああああああああ
きんっきに冷えてやがるぜ

雪音さん、おやすみなさい… 🌙 zzz

配信楽しかったです～新クリーチャーっ！！

- 「ああああ」「… 🌙 zzz」「ナリよ」など、無数にある文末表現をすべて人手でタグ付けするのは不可能
- 教師なし学習？
 - テキストの最後は、必ず文末; 最初は、必ず文頭
 - 統計的なヒントは豊富 → できるはず
- 先行研究：Punkt (Kiss+ CL2006), Where's the Point (Minixhofer+ ACL2023)などは、欧米語のピリオドや行末に依存したヒューリスティック

LLMによる文分割

- 現在のSOTA:
“Segment Any Text (SAT)” (Frohmann+ ACL 2024)
- LLMに次の(ひどい)プロンプトを与えてテキストを文に分割

General LLM Prompt

Separate the following text into sentences by adding a newline between each sentence. Do not modify the text in any way and keep the exact ordering of words! If you modify it, remove or add anything, you get fined \$1000 per word. Provide a concise answer without any introduction. Indicate sentence boundaries only via a single newline, no more than this!

LLMがテキストを勝手に書き換えることもあるため、後処理が不可欠



こんないい加減な方法がSOTAとは。。

セミマルコフモデルによる定式化

$s =$

お	は	よ	ー	今	家	?	そ	う	い	え	ば
---	---	---	---	---	---	---	---	---	---	---	---

 $\cdots$

$\mathbf{b} =$

1	0	0	0	1	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

 $\cdots$

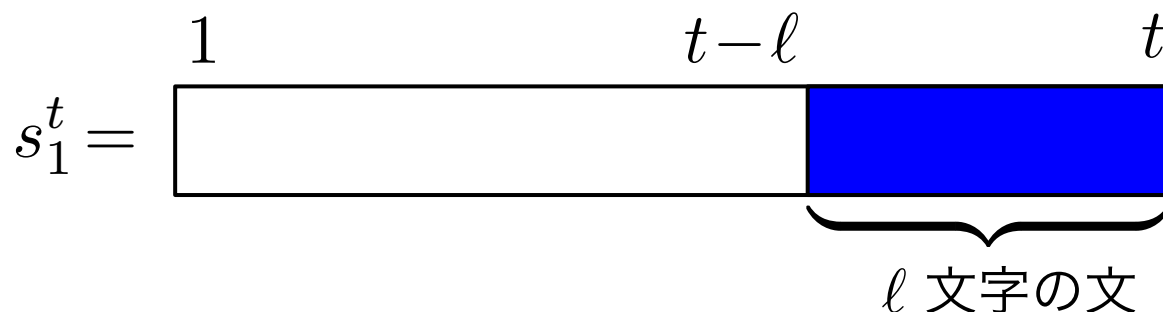
- テキスト $s = c_1 c_2 c_3 \cdots c_T$ の各文字の背後に、文がそこから始まるかを表す二値の潜在変数 $\mathbf{b} = b_1 b_2 b_3 \cdots b_T$ があると仮定 ($b_t = 1$ の場所が文頭)
- 観測データの確率

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

を最大化する $\mathbf{b}$ を学習する

– 探索空間は 2^T で膨大にある

動的計画法による文分割

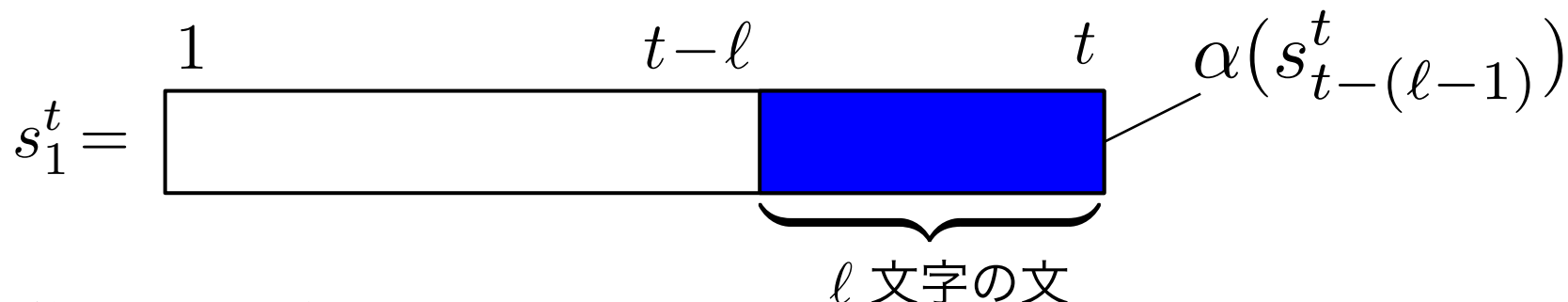


- 前向き変数 $\alpha(s_{t-(\ell-1)}^t)$ を、「時刻 t までの文字列で最後の ℓ 文字が文になっている周辺確率」とする
- 定義から、 $\alpha(s_{t-(\ell-1)}^t)$ は次のように展開できる

$$\alpha(s_{t-(\ell-1)}^t) = \underbrace{p(s_{t-(\ell-1)}^t | b_t = 1, \sim)}_{\text{文の確率}} q(1-q)^{\ell-1} \sum_{j=1}^{t-\ell} \alpha(s_{t-\ell-(j-1)}^{t-\ell})$$

($q = p(b_t = 1)$: 分割の事前確率)

動的計画法による文分割 (2)



- 導出は定義から以下：

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

- $\sim$ は、 $b_{t-1} = \cdots = b_{t-(\ell-1)} = 0$ を表す

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

「文の確率」の計算

- 文の確率

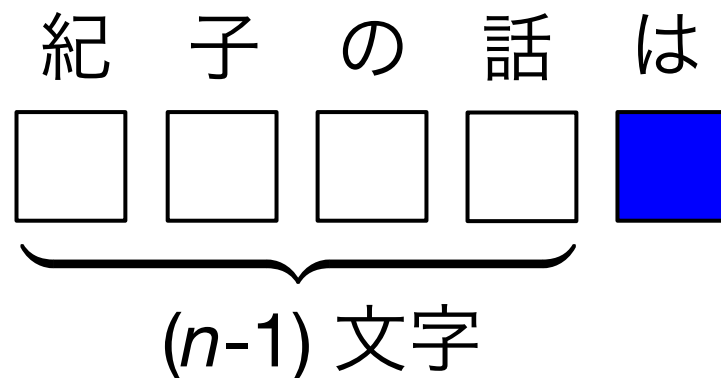
$$p(s_{t-(\ell-1)}^t | b_t = 1, \sim) = \prod_{j=1}^{\ell} p(c_{t-\ell+j} | h_{t-\ell+j-1}) \cdot p(\$ | h_t)$$

は、文字 n グラムモデルを用いて計算できる

- h_t : 時刻 t までの n グラム文脈 (文頭文字 $^$ を含む)
- $\$$: 文末を表す特殊文字

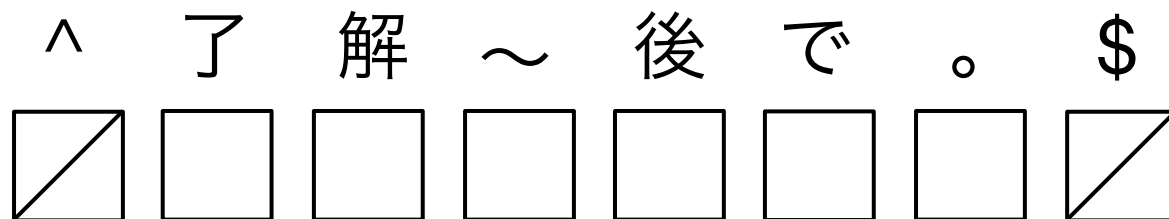
- 学習データの文分割に伴って、 n グラム言語モデルも更新される
(=何が文頭や文末になりやすいか、が更新される)

n グラムモデルとは



- 前の $(n-1)$ 文字から、頻度に基づいて次の文字を予測するモデル (単語ベースでもよい)
 - 連続した n 文字の頻度を数える= n グラムモデル
- n が大きくなると頻度がスパースになるので、 $n \leq 5$ くらいが普通 (cf. ∞ -gram model: 持橋2007)
- 本研究では、階層Pitman-Yor過程に基づくベイズ n グラム言語モデル(Teh+ 2006)を使用 ($n=5$)

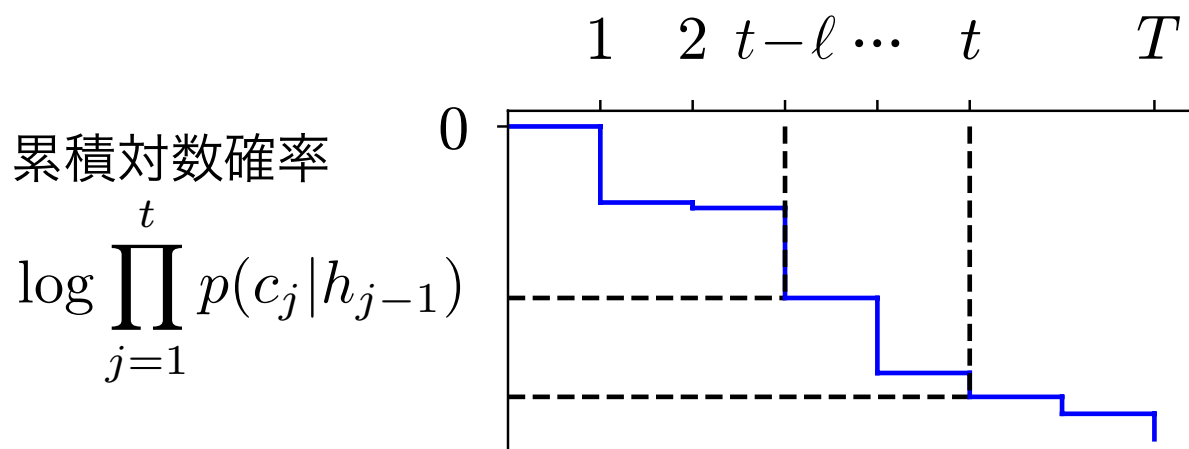
BOSとEOS



- n グラムの頻度を数える際には、実際には
文頭と文末を表す特殊文字 $\wedge$, $\$$ を追加して数える
→ どんな文字列が文頭や文末になりやすいかがモデル化できる
 - 学習の際は、文分割に従って $\wedge$, $\$$ を内部に追加
 - 実装上は、 $\wedge$ と $\$$ は同じでよい (たとえばID=0)
- 2グラムの場合、上の例では
“ $\wedge$ 了”, “了 解”, “解 ～”, ..., “で 。”, “。 $\$$ ” の頻度を++

MCMC法による文分割の学習

- 前向きフィルタリング–後向きサンプリング法 (FFBS) を用いて、学習テキストの文分割をサンプリング
→ 言語モデルを更新、を繰り返す
- 文のnグラム確率は、累積確率 $\Phi(t) = \sum_{j=1}^t \log p(c_j | h_{j-1})$ を事前に計算しておけば、 $\Phi(t) - \Phi(t-\ell)$ を使って $O(1)$ で求められる → $O(T^3)$ から $O(T^2)$ に学習が高速化



全体の学習アルゴリズム

- 全体がGibbsサンプラーになっている

```
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  $n = \text{randperm}(1 \cdots N)$  do
4:     if  $j > 1$  then
5:       現在の文分割でのテキスト  $s_n$  の  $n$  グラム頻度
        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  $n$  グラム頻度を
        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.

改行と事前確率

- 本研究の定式化で、分割の事前確率 $q = p(b_t = 1)$ は重要
 - q を正しく扱うことで、文分割の性能が大きく改善した
 - 元のテキストで改行があった部分は、文末である可能性が高い → 事前確率 q が異なるはず
 - q_2 : 改行があった位置
 - q_1 : 句読点があった位置
 - q_0 : それ以外の位置
- の3種類の事前確率を導入し、 q_i の事後確率分布を学習中に求めて q_i 自体もサンプリングして学習

実験

- 学習データ：
 - APIから取得したランダムな日本語ツイート200K個
 - 国語研日本語書き言葉コーパスのCore部からランダムな200K文
(半教師あり学習の場合)
- テストデータ
 - Twitter (Xのこと)：ランダムな894ツイートをクラウドソーシングで文分割 (5名, 一致率89.6%)
→ 一致率>80%, =100%に分けて評価
 - BCCWJ：
10K文をランダムに10文ずつ連結したテキスト
- 言語モデル：5グラムの文字HPYLM (ベイズ n グラム)

文分割結果 (1/2)

- 元ツイート：

閃光のハサウェイを
観に行こうとしたけど
ヤメー

絶対数はともかく容易に想像できる
...(=) トオイメ
それで逆に別姓くんが皆から距離を置かれて「いじめ」に発展する可能性もある

いとちゃんおはよ〜♥(・ω・๓)
ギリギリセーフだねw
声がガラガラになるってことは
歌ってみたの録音してるとか?(-v-)ニヤ
(°Д°)アッ!?
また21位って書いてあるぞwww

わぁ読破ありがとうございます……! 😊 ✨ (もしや先ほどの方でしょうか?違ってたら\
ごめんなさい💧) 共通するものがあるって嬉しいです! 児童書版は本当健やかであれ!\
|ですね♪
質問の件ですが(リプ続)

文分割結果 (2/2)

- 提案手法による文分割結果：

閃光のハサウェイを観に行こうとしたけどヤメー

絶対数はともかく容易に想像できる...(==) トオイメ
それで逆に別姓くんが皆から距離を置かれて「いじめ」に発展する可能性も在る

いとちゃんおはよ〜♥(・ω・๓)ギリギリセーフだねw
声がガラガラになるってことは歌ってみたの録音してるとか?(-v-)ニヤ(°Д°)アッ!?
また21位って書いてあるぞwww

わぁ読破ありがとうございます……! 😊 ✨
(もしや先ほどの方でしょうか?
違ってたらごめんなさい💧)
共通するものがあって嬉しいです!
児童書版は本当健やかであれ!
ですね♪
質問の件ですが(リプ続)

実験結果 (Twitter: 崩れたテキスト)

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	<u>86.6</u>
	ベイズ半教師あり学習	83.3	93.1	<u>87.9</u>

- LLMによるアドホックな文分割を超えて、**提案手法が世界最高精度**
 - 教師あり学習のBunkaiや, 自己教師あり学習のWtP (ACL 2023)より高性能
 - 標準的な文を事前学習した半教師あり学習で, さらに改善

実験結果 (BCCWJ: 標準的テキスト)

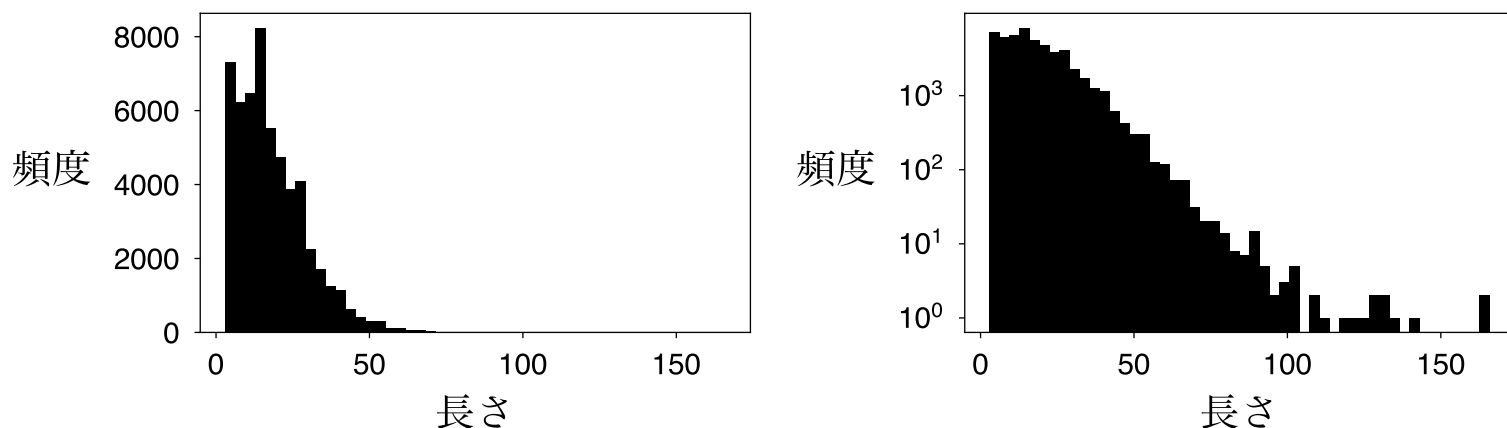
モデル	説明	精度	再現率	F_1
PySBD [2]	ルールベース	66.1	54.3	59.6
Ersatz [1]	Transformer	63.2	44.0	51.9
WtP [4]	自己教師あり学習	84.5	71.6	77.5
SAT [5]	大規模言語モデル	80.0	74.6	77.2
USBD	ベイズ半教師あり学習	94.7	88.7	<u>91.6</u>

- 標準的な文の文分割についても、提案法は最高精度
 - 文をランダムに繋いだテストデータなので、結構難しいタスク
- きちんと統計的な定式化をすることが有効

まとめと展望

- テキストの文分割の完全な教師なし学習を提案
 - セミマルコフモデルによる統計的な定式化とMCMC
 - LLMによる文分割を超えて、現在世界最高精度
- ツイートなどテキスト末尾/先頭の情報の統計を利用
 - ヒューリスティックなし; 日本語/中国語/タイ語等にも適用可能
 - 教師なし単語分割との違い→ 二重分節ではない、探索空間の圧倒的な広さ ($O(T^3)$ だが高速化して $O(T^2)$)
- 今後
 - 言語モデルのニューラル化 (現在は5gram)

おまけ: 文の長さの統計モデル



(a) 横軸が文長, 縦軸が頻度を表します. (b) 縦軸を対数軸で表した場合.

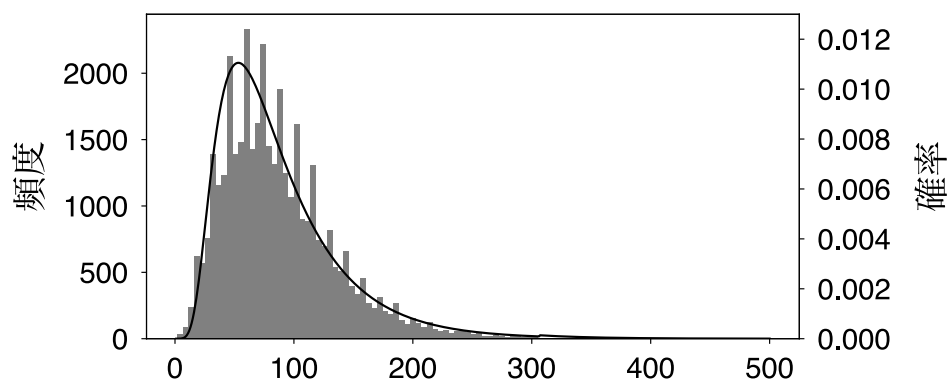
図 4.8: Brown コーパスでの文の長さ (単語数) の分布.

- Brownコーパス (英語100万語の均衡コーパス) での文の長さの分布
 - 横軸は文長、縦軸は左は頻度、右は頻度の対数
- 接続詞で繋がった、非常に長い文もかなり存在

文の長さの統計モデル (2)

- Sichel分布 (Sichel 1974, JRSSA) による当てはめ

$$p(n|\theta, \alpha, \gamma) = \frac{\sqrt{1-\theta}^\gamma}{K_\gamma(\alpha\sqrt{1-\theta})} \frac{(\alpha\theta/2)^n}{n!} K_{n+\gamma}(\alpha)$$



推定したパラメータ:

$$\xi = 87.23$$

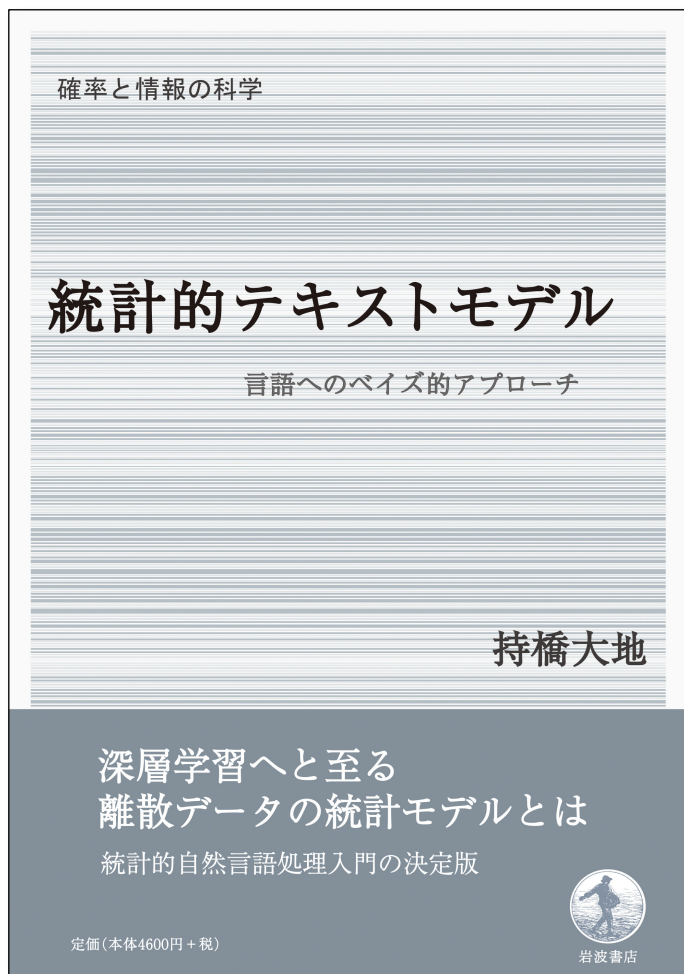
$$\alpha = 22.69$$

$$\left(\xi = \alpha\theta / (2\sqrt{1-\theta}) \right)$$

図 4.9: 京大コーパスの文長の分布とポアソン逆ガンマ分布による当てはめ. 分布の凹凸は, 全角文字の長さを 2, 半角文字を 1 として数えているためです.

— もっと新しい研究もあります (Borbély and Kornai 2019)

『統計的テキストモデル』



- この話もコラムとして含まれた、私の教科書が発売中です (岩波書店より2025/7発売)
- 久保さん緑本と同じシリーズ
- 全体的に、非常にベイズ的だと思います
- 紹介・サポートページ:
<http://chasen.org/~daiti-m/textmodel/>