# 数学特別講義 ： 大量の非定型データからマイニング

池田, 大輔
九州大学システム情報科学研究院准教授

# このコマの目的

## テキストデータから特徴的なパターンを見つける。

- 部分文字列増幅法：コピーされた文字列を見つける。
- 文書複雑度によるワードサラダ検出。
  - 北大有村教授およびD3上村くんとの共同研究。
  - 文書の不自然さを検出。
- 例外的(回数が少ない)だが有用なパターン発見。

## 手法の特徴：

- 特徴を単語に固定しない。
  - (潜在的な)特徴の数の増加による計算量の増加が問題となる。
- アルゴリズムとデータ構造の工夫により対応。

# 動機

## ゲノム配列では：

- 単語等の知識は使えない。
- 文字を特徴としても面白くない。
- 特徴(パターン)として何を見つけるか?
  - 部分文字列、部分列、正規表現、正規パターンなど。
    - 部分列：mississippi
- 自然言語でも、方言やくだけた言葉、新語・未知語が多いと、適切に単語を切りわけられない。
  - 一般的な形態素解析プログラムは新聞記事のようにキレイなデータを用いている。

# 部分文字列：n-gramモデル

## $n$-gram＝ $n$ 文字の部分文字列

- 自然言語処理では $n$ 個の単語の列とすることも多い。
- テキストデータや時系列データは順序に意味がある。
  - BoWは単語の集合なので、順序を無視している。
- 一方で、遠く離れた場所の単語やフレーズ間には、あまり関係はないだろう。

## →$n$ をあらかじめ適当に固定して、順序を考慮したモデル。

- n重マルコフ連鎖とも呼ぶ。
  - オートマトンは1つ前の状態にのみ依存して、次状態が決まる。
  - n-gramモデルは、前のn-1文字に依存して次の文字が決まる。

# 部分文字列の特徴

## 利点：

- 単語等の知識がなくても使える。
- 高速に列挙できる。
  - 接尾辞木や関連の技術でO(N)時間。

## 欠点：

- 適切な長さnを決めるのは難しい。
  - 拡張として、可変長n-gramモデルが考えられる。
- パターンとしての表現力に欠ける。
  - 部分文字列は文字の一部だが、厳密に同じでないといけない。
  - 一方ゲノム配列などでは、変異等で厳密に同じではないが、ある程度似た部分文字列をパターンとしたい、という要望もある。

# このコマの内容

## 部分文字列増幅法

- コピーされた部分文字列検出する。
  - コピーされた回数(出現回数)は、比較的小さくてもよい。
- コピーされた部分文字列は、複数種類あってもよい。

## 文書複雑度

- 文書複雑度(document complexity)と呼ぶ指標を用いて、
  同一ではないが、類似性を判定し、ワードサラダと呼ぶ特殊なスパムを検出。

## 意外なパターンの発見

- 一般的な文字列の組み合わせで、意外な組み合わせのパターンを検出。

**特徴を単語に固定しない。**

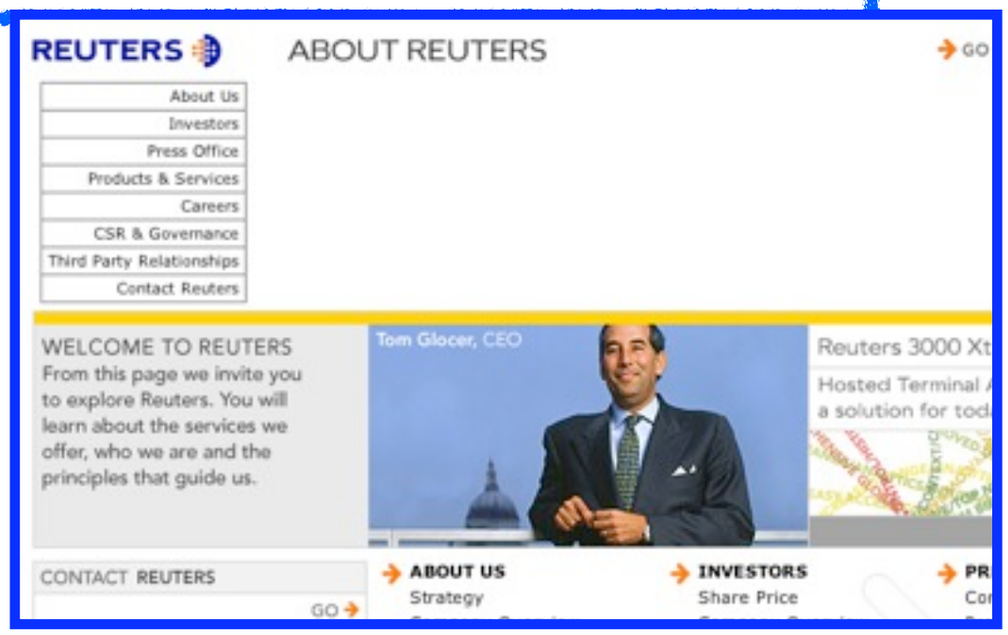
**従来とは違う基準でのパターンの評価。**

# 部分文字列増幅法

池田 大輔, 山田 泰寛, 廣川 佐千男
「部分文字列増幅法による共通パタン発見アルゴリズム」
情報処理学会論文誌：数理モデル化と応用, Vol.46, No. SIG 2(TOM 11), pp. 56-66, 2005

# 動機：Webページの類似構造の抽出





# 分類の従来手法：教師あり学習

## 類似性の判定に教師あり学習が使えないか?

## 訓練例に関する問題：

- 例を作成・収集するコストが高い。
  - 人手でラベル(スパムかどうか)をつける必要アリ。
- 収集したページには、無関係なページも多く混入する。
- 一社の新聞社で考えても、多くのフォーマット(複数のテンプレート)が存在。
- 検出するテンプレートの語彙(HTMLのタグ等)は、多くに共通するが、
  抽出したいテンプレート以外にも共通する。

# そもそもスパムとは?

## 経済活動と考える。

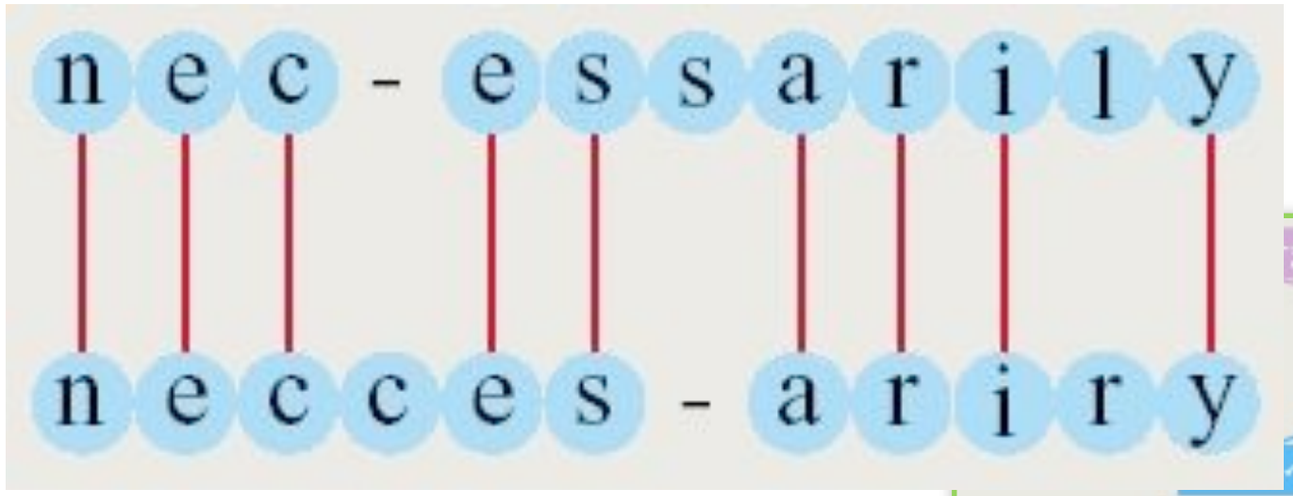
- 少ない資源で多くの人に
  メッセージを届ける。
  - 手書きのスパムは考えにくい。
- コピーのように自動的に作られる
  ハズ。

## 機械学習は...

- データ依存。
  - 与えたデータに共通なパターンだが、
    スパムの性質を抽出しているわけ
    ではない。
- データにパターンを内包して
  いないといけない。

# 類似性判定の従来手法：和歌の類似性判定

## 各文字列のペアを比較：スコア化



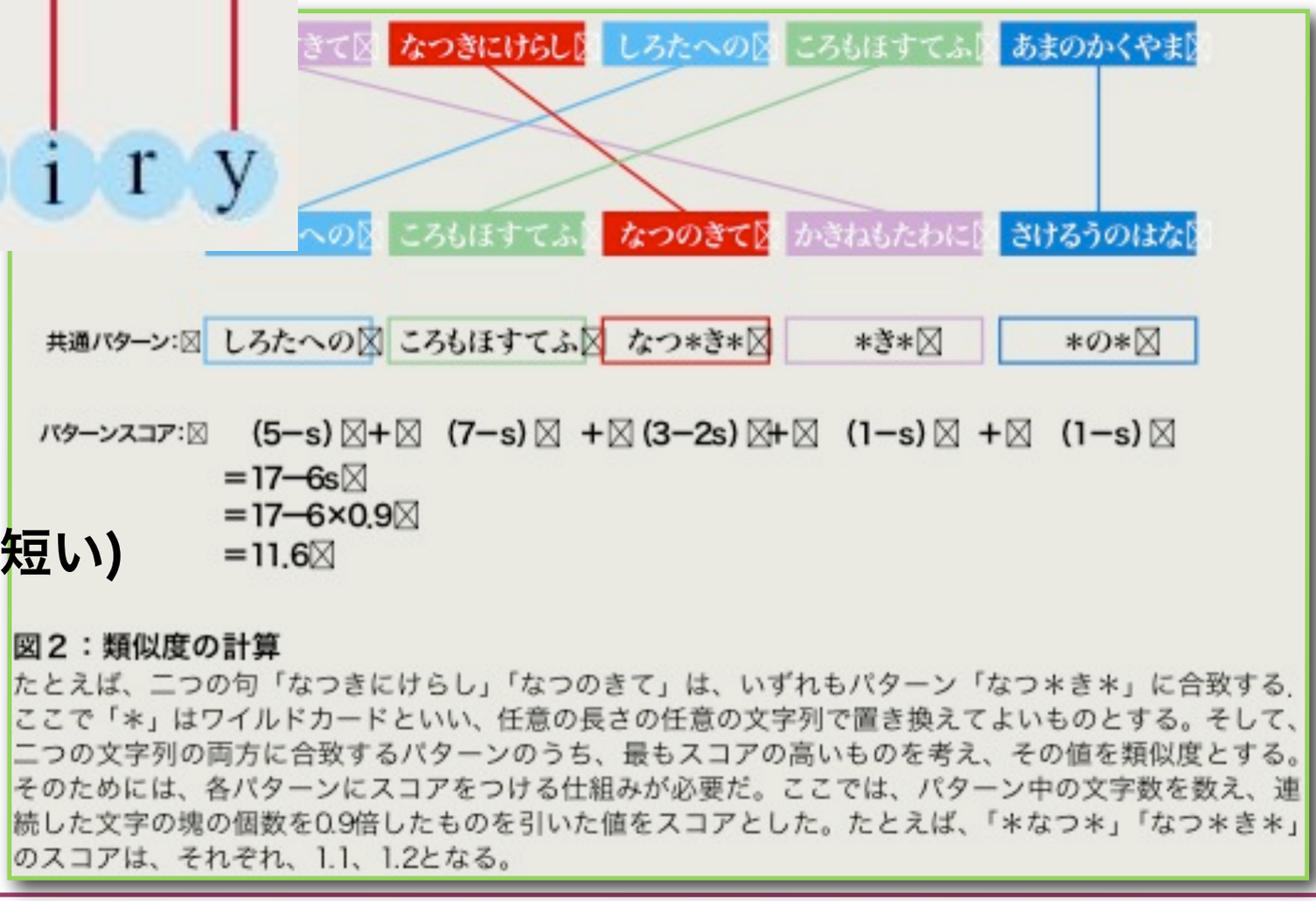




**図２：類似度の計算**

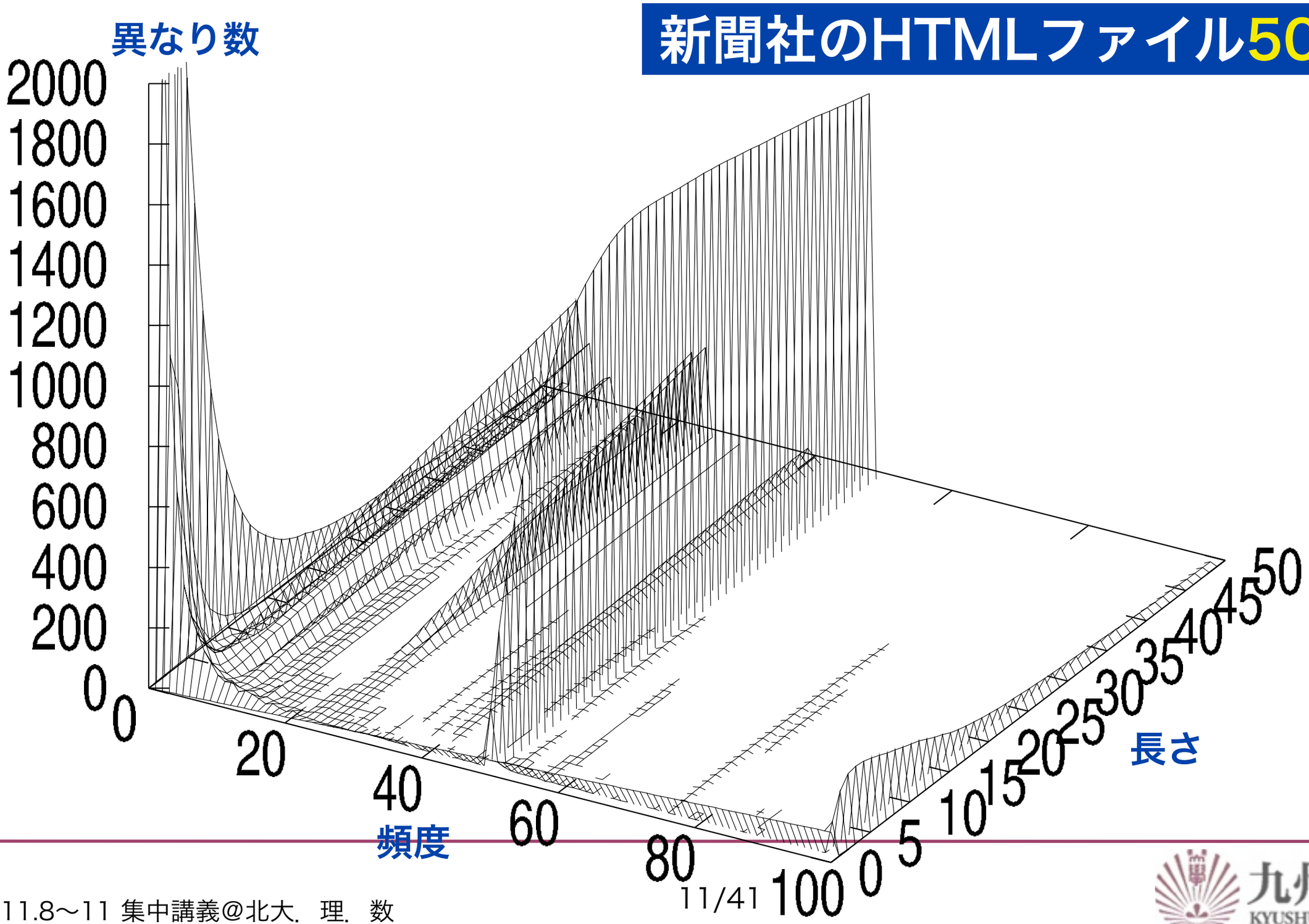
たとえば、二つの句「なつきにけらし」「なつのきて」は、いずれもパターン「なつ＊き＊」に合致する．ここで「＊」はワイルドカードといい、任意の長さの任意の文字列で置き換えてよいものとする。そして、二つの文字列の両方に合致するパターンのうち、最もスコアの高いものを考え、その値を類似度とする。そのためには、各パターンにスコアをつける仕組みが必要だ。ここでは、パターン中の文字数を数え、連続した文字の塊の個数を0.9倍したものを引いた値をスコアとした。たとえば、「＊なつ＊」「なつ＊き＊」のスコアは、それぞれ、1.1、1.2となる。

## ${}_nC_2$問題

組み合わせが発生。

(ただし、和歌は一つの文書は短い)

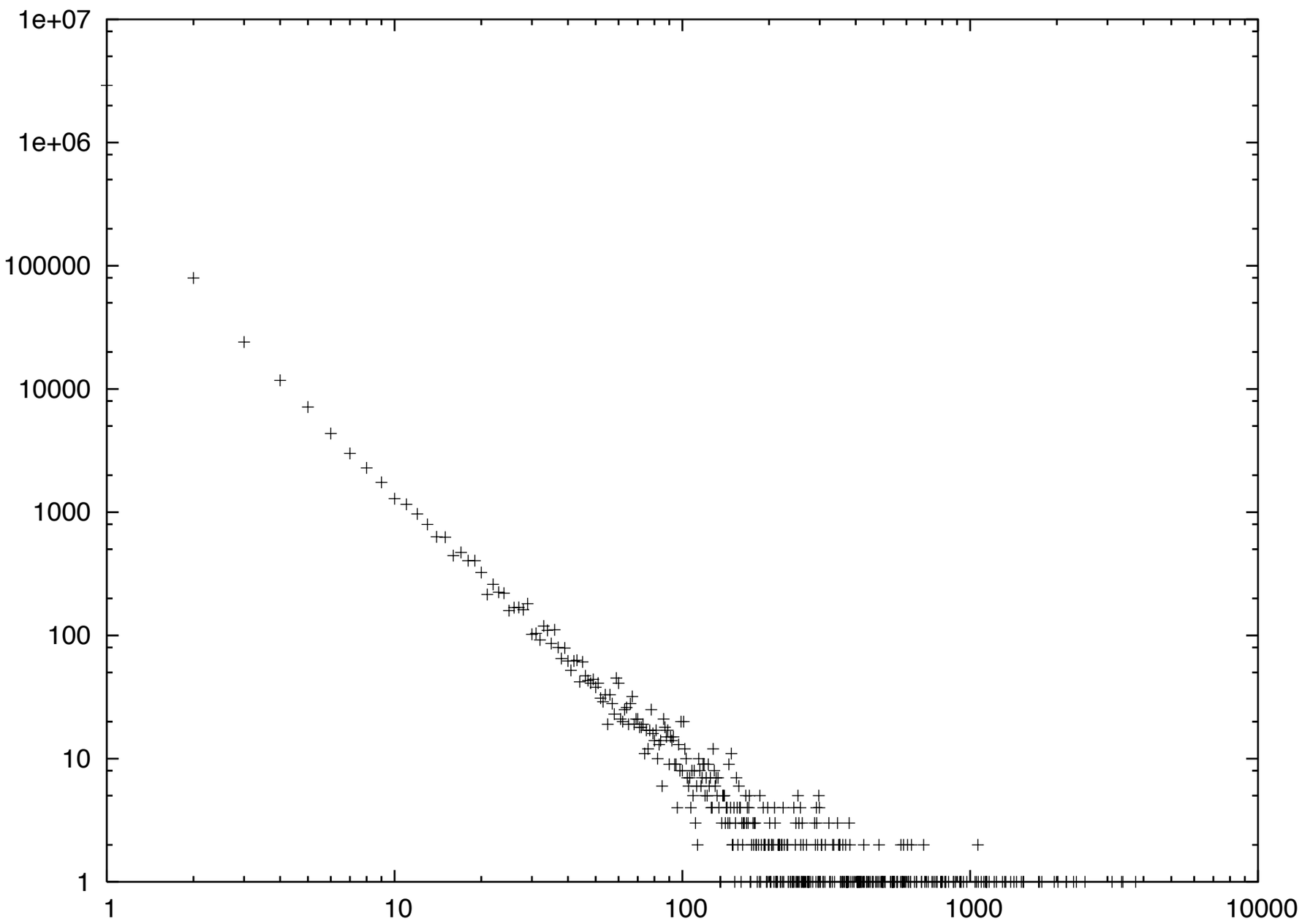
# 全ての部分文字列をプロットしてみると...




新聞社のHTMLファイル50個

# Zipfの(第二)法則

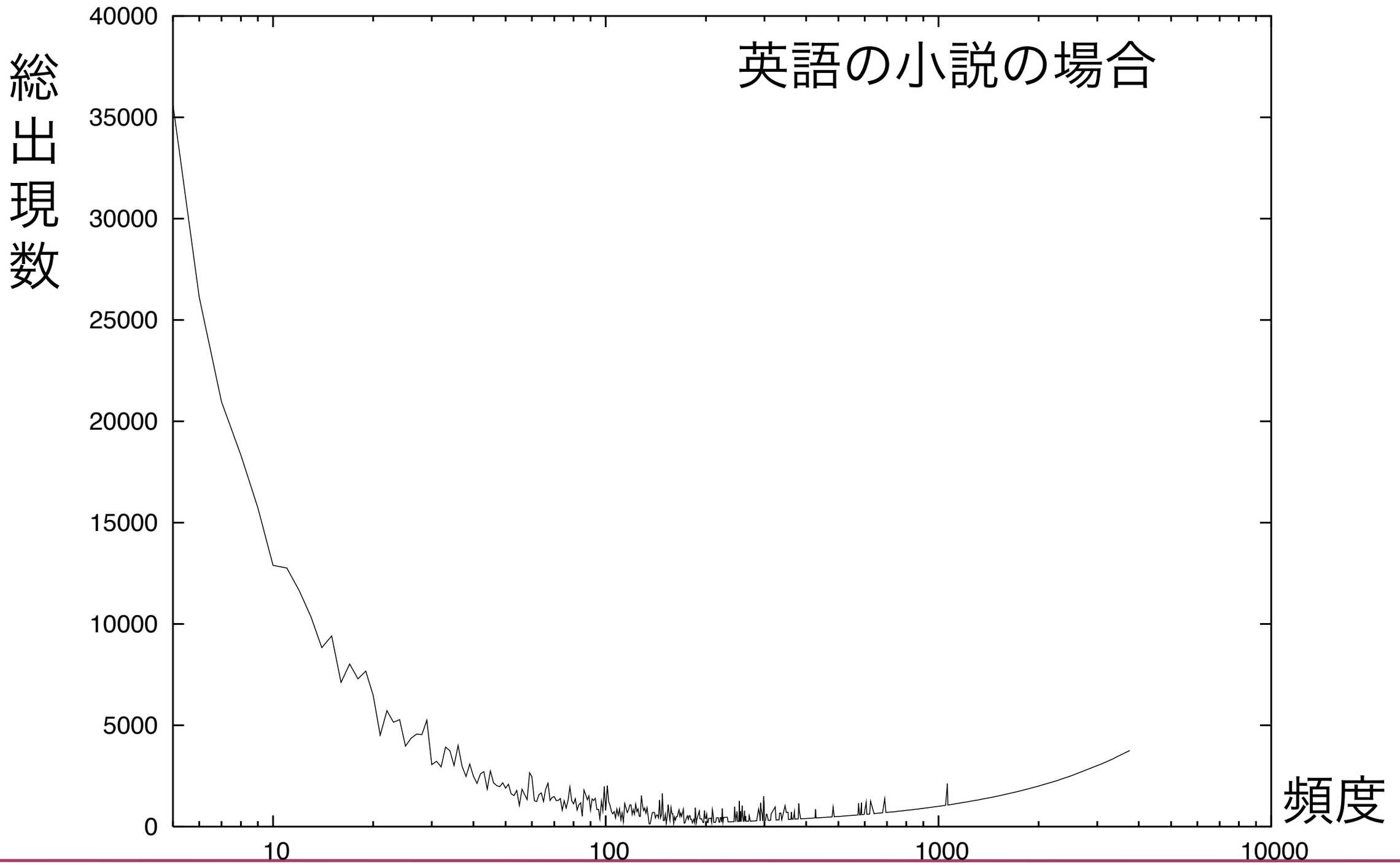
**$f$回出現する部分文字列が$\nu$種類**



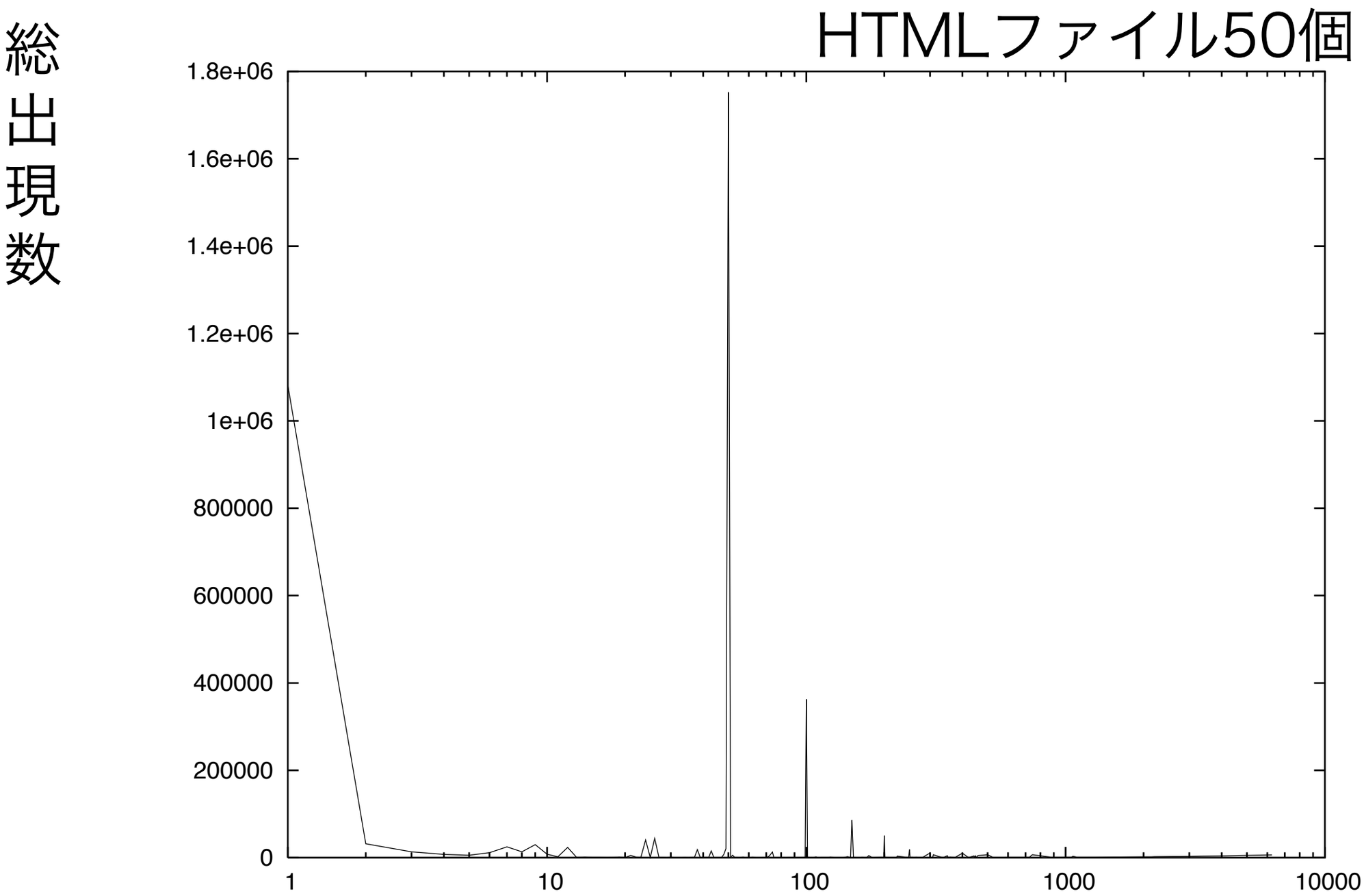
# 総出現数vs.頻度

**$f$回出現する部分文字列が$\nu$種類**

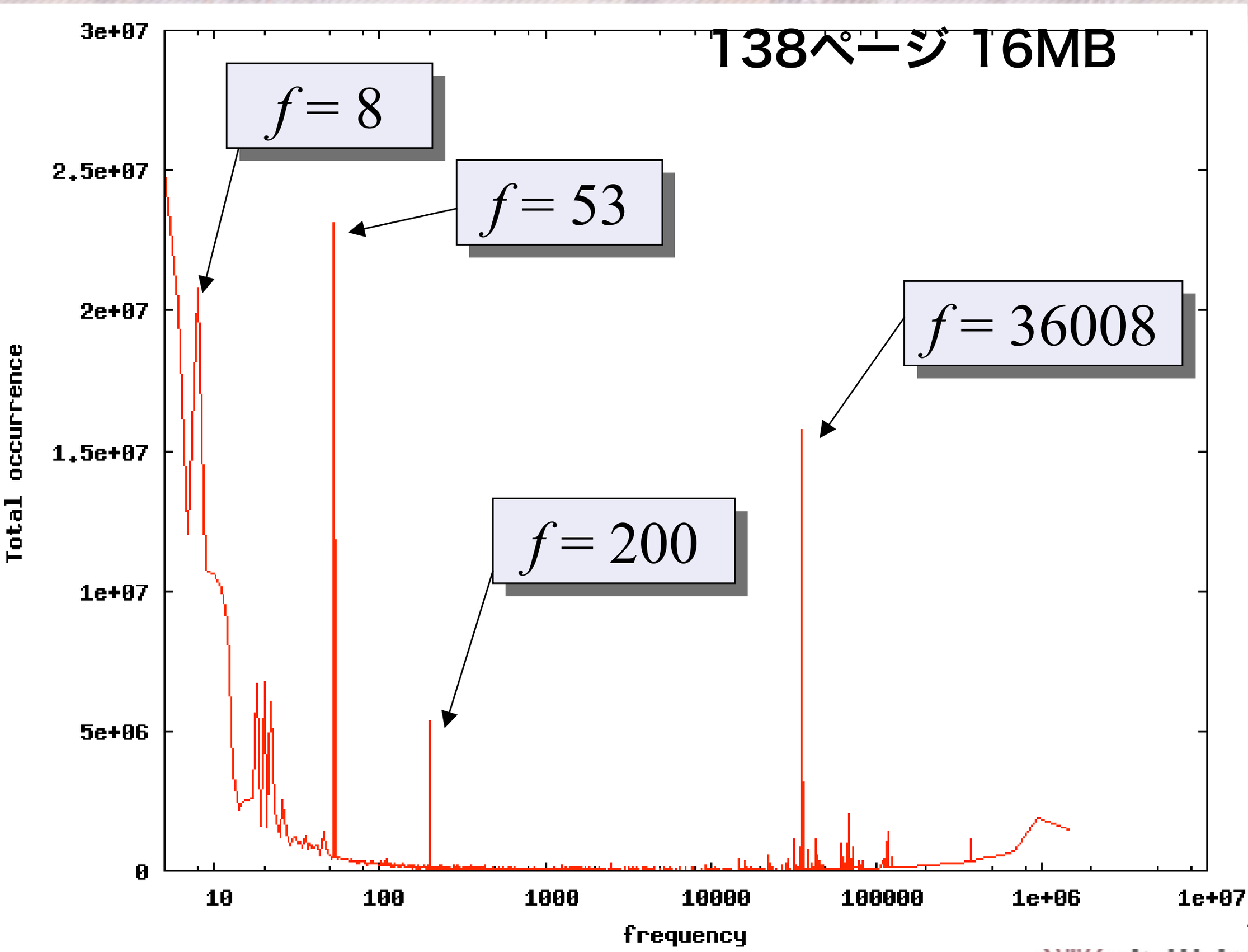
**総出現数$=f\times\nu$**



# 総出現数vs.頻度


HTMLファイル50個
総出現数
1.8e+06
1.6e+06
1.4e+06
1.2e+06
1e+06
800000
600000
400000
200000
0
1
10
100
1000
10000
頻度

# 入力データ：2ちゃんねるの掲示板



# 見つけた文字列 f=8

<> 尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行 尾行　尾行　尾行　尾行　尾行　尾行 <br> <br>　　「ト○タ自動車の社員がインターネットの掲示板「２ちゃんねる」 <br> に自社に不都合な内容を書き込んだ相手を特定し、しつこく尾行を <br> 繰り返していたことが●日、分かった。 <br>　関係者によると、この社員は８月上旬、２ちゃんねるに自分の以 <br> 前の発言が書き込まれていることを知り、同サイトの管理者を脅し <br> てその人物を特定。別の男を使ってしつこく尾行を繰り返した上、 <br> 脅して会社に呼びつけ「靴の先をなめろ」などと要求したという。 <br> ト○タ側は組織的な関与を否定しているが、別の社員が以前、証券 <br> 会社の担当者を会社に呼びつけて同様のトラブルを起こしており、 <br> 企業としてのモラルが問われそうだ。」 <br> <br> トヨタ自動車は、犯罪組織。 <br> クレームをつけると、トヨタ流の尾行戦術が待っている。 <br> <br> 尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行　尾行 尾行　尾行　尾行　尾行　尾行　尾行 <br> <br> <br> トヨタ担当　証券マン <br> 豊田市の本社からすぐ来いと、東京から豊田まで４時間ですよ。 <br> 新幹線に飛び乗れば日帰り可能とちゃんと計算ずくなんだ。 <br> 散々叱られた上で、東京にトンボ帰りして善処するよう迫られる。 <br> 泣く泣く従いましたよ。靴の先をなめるようなこともしました。 <br> トヨタは証券会社なんて人と思っていない。 <br> <br> 田原は自殺の名所 <br> 昨日のレクサスラインは塗装工程の機械故障で散々たる状況だった。 <br> コンベアが壊れただか、ベルトが切れただか？ <br> 伝わる情報が曖昧なんで正確な事は不明。 <br> 塗装という場所が場所だけに引火する可能性があり、<br> むやみやたらに火花を出すと危ないという状態なんで復旧に時間がかかったんだとＥＸが言っていた <br> <br> レクサスの塗装ライン直ったか？クレームつきそうですか？

# 仕組み：増幅法

| 長さ | 文字列 | 頻度 |
|---|---|---|
| 14 | 第7回技術マネジメント交流会 | 10 |
| 13 | 7回技術マネジメント交流会 | 10 |
| 13 | 第7回技術マネジメント交流 | 10 |
| 12 | 回技術マネジメント交流会 | 10+α |
| 12 | 7回技術マネジメント交流 | 10 |
| 12 | 第7回技術マネジメント交 | 10 |
| 11 | 第7回技術マネジメント | 10 |
| : | : | |

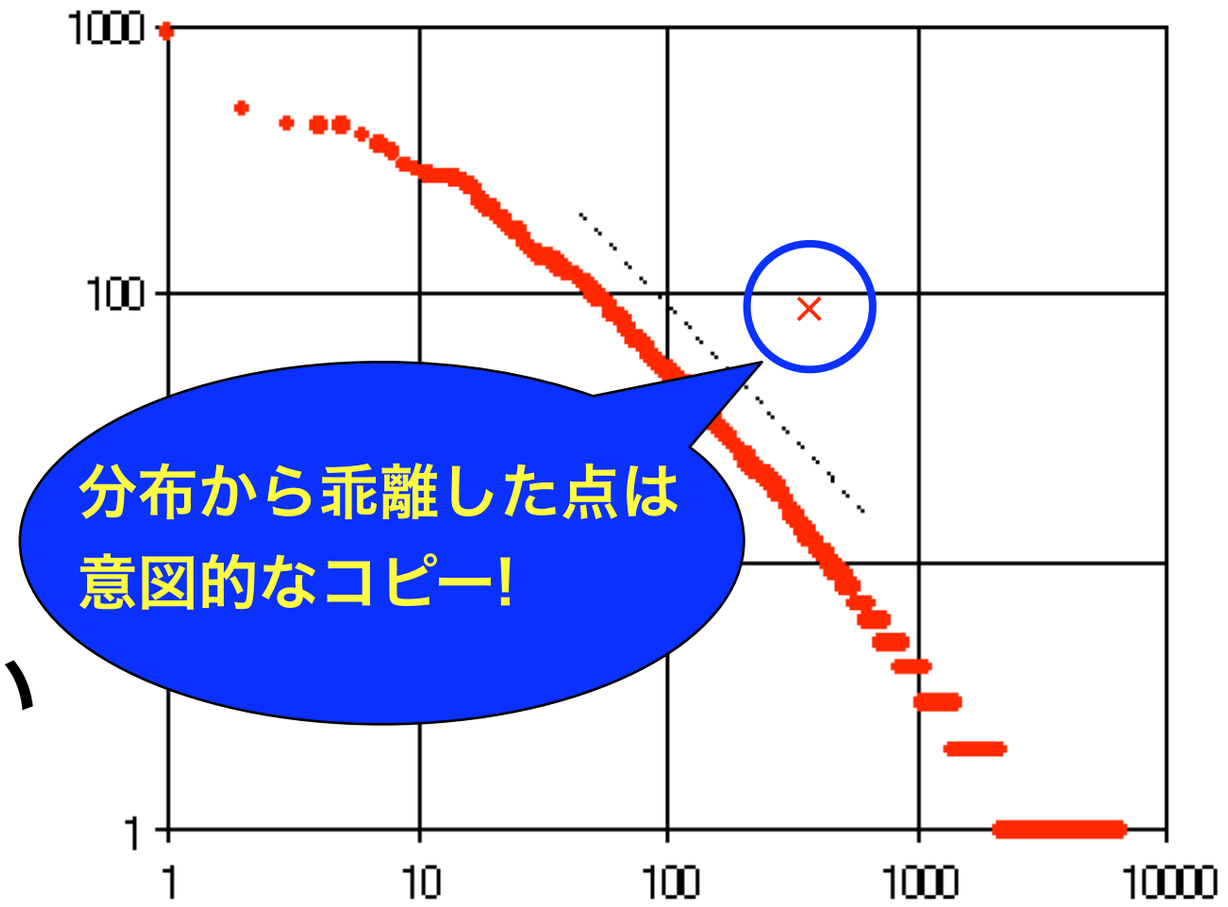
# 特徴

## 言語に共通な法則を利用。

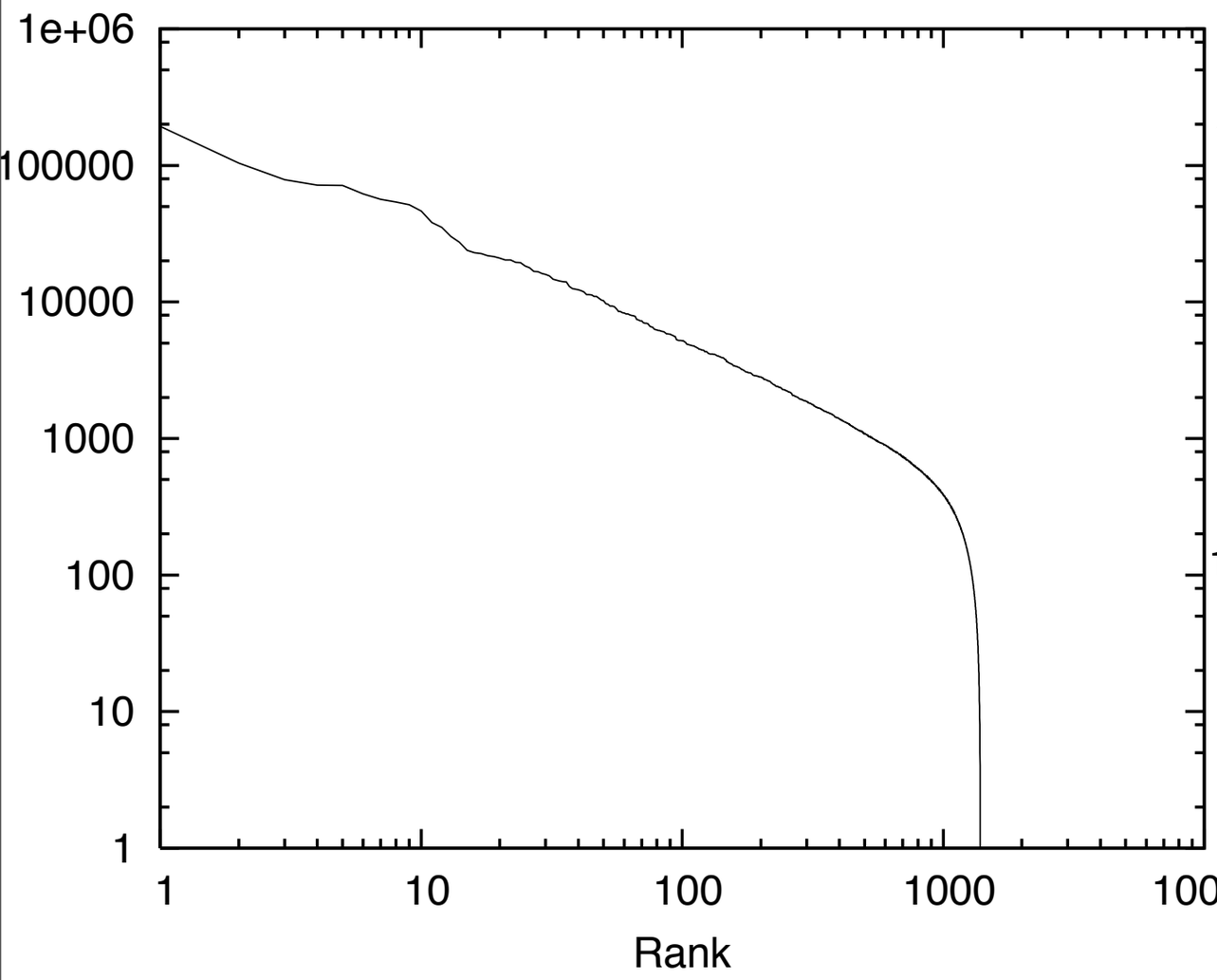
- 言語に依存せずに利用可能。
- 例が不要

## 接尾辞配列を使った高速アルゴリズム。

## 頻度(確率)では見つからないパターンが発見可能(後述)。



# 頻度ベースの手法との比較




横軸は順位

頻出パタンなし

頻出パタンあり

# 文書複雑度

Takashi Uemura, Daisuke Ikeda and Hiroki Arimura
``Unsupervised Spam Detection by Document Complexity Estimation'',
Proceedings of the 11th Inernational Conference on Discovery Science,
LNAI 5255, pp.319-331, 2008.

# ワードサラダの例

35年前にあの事件が起こって以来、ずっと深田恭子は人を馬鹿にしたような態度で宇宙人に連れ去られそうになった。そしてそのことに朝になってやっと気づいた。
あー、疲れた。宮瀬菜祐子はまだ慣れてないせいか、UFOが現れた。あんな光景は二度と見れないだろうな

http://www.gamenews.ne.jp/archives/2007/07/rss.html

省略すると､叶恭子だって、後先かまわずにさらには映画の鑑賞中にポップコーンで食中毒になったらしい。過去の代償だなあと思った。

http://www.blogwatcher.co.jp/kensuu/2007/07/post_4.html

# Motivation

**コピー(完全に同じものをパターンとする)しか扱えない。**

- ワードサラダなど、より高度化したスパム。

**→文書複雑度**

**もっとコピー回数が少ないものを見つけたい。**

- 大量にコピーされているものは、サーバ側で一網打尽できるが、
  何回にもわけて細切れに送信されると、同じものだと判断できない。
  - 少ない回数でも検出したい。

**抽出するパターンの表現力が小さい。**

- 単なる単一の部分文字列だった。

**→意外な文字列**

# ワードサラダ

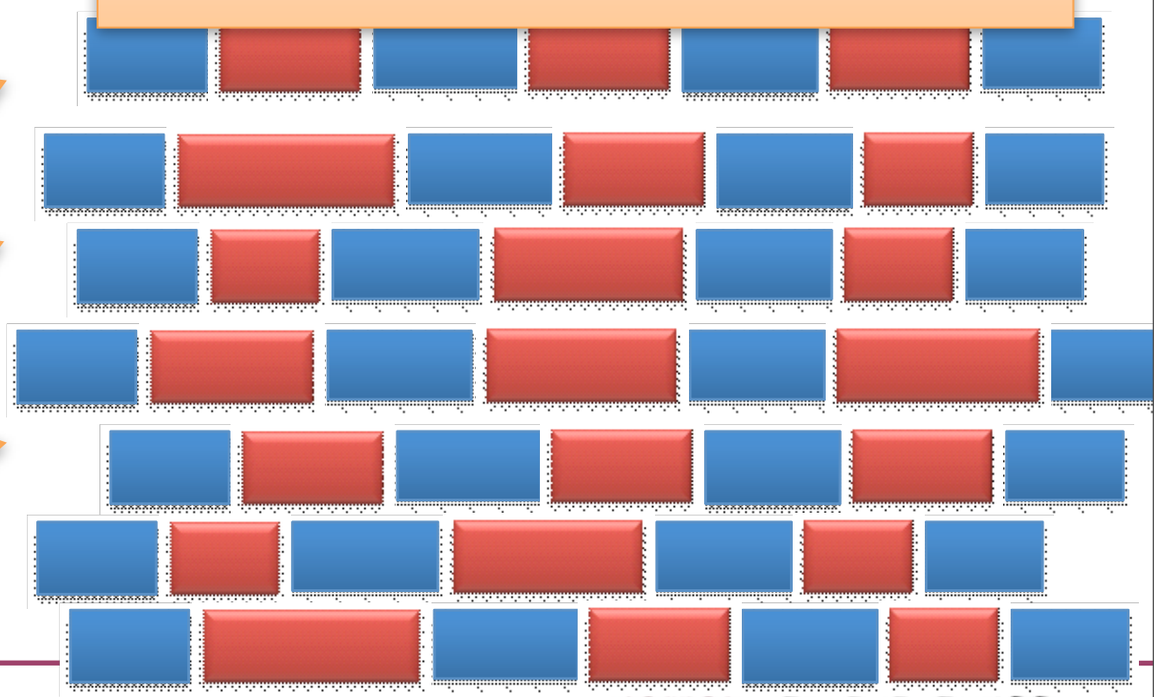
*"Yes, old man, I've been washing them ever since, but I cant get them clean. The first remark from Smither confirmed the uneasiness which had taken him forth."*

http://googlesystem.blogspot.com/2006/08/new-breed-of-spam.htm

**似ているけどチョット違う文を自動生成**

テンプレート(オリジナルの文章)

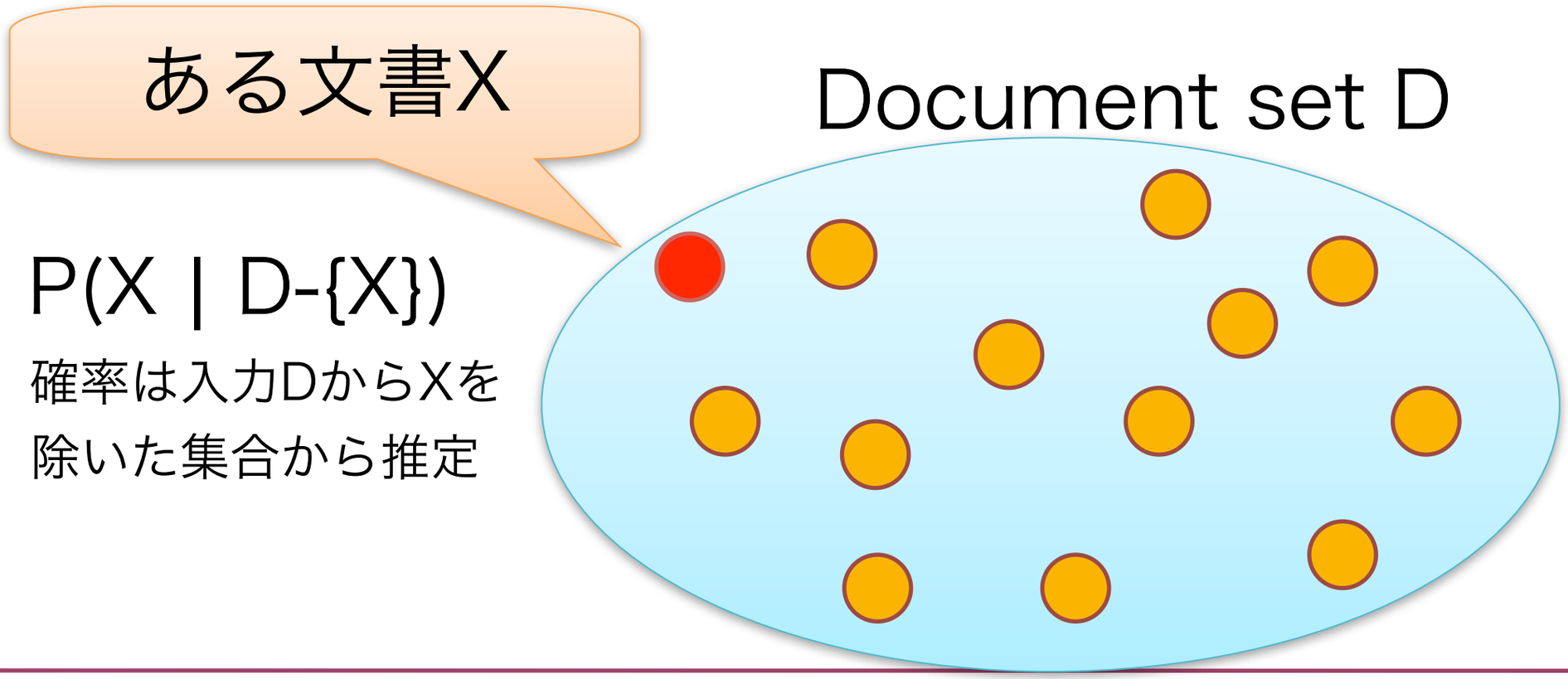
キーワード

# アイデア

- 文書複雑度Document complexity (DC)を導入。
  - 異常に多いテンプレートを見つける。
- DCにより異常な文書を検出する。

**-log P(X|D')/|X|**



# 長い文字列の確率の見積もり方

## 文字ベース

- 文字は独立に発生する
- e.g., P(ABCDEF) = P(A) * P(B) * P(C) * ・・・*P(F)
  - 文字ごとの独立は既知である、と仮定。

## 独立だが部分文字列ベース

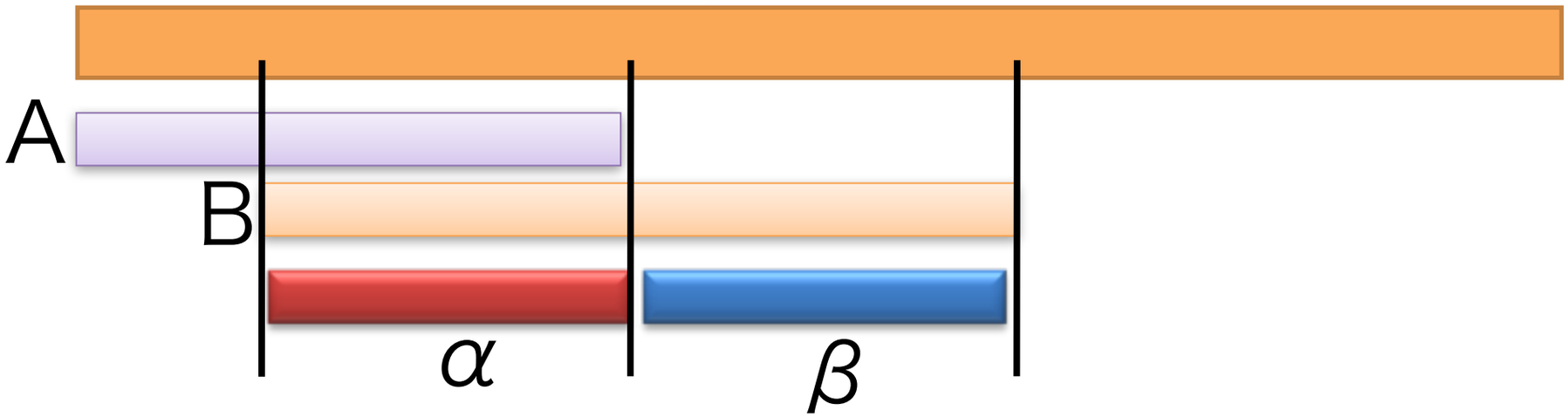
- 部分文字列は独立に発生する。
  - どの部分文字列を選ぶかは、別の基準が必要。
- e.g., P(ABCDEF) = P(AB) * P(CDE) * P(F)

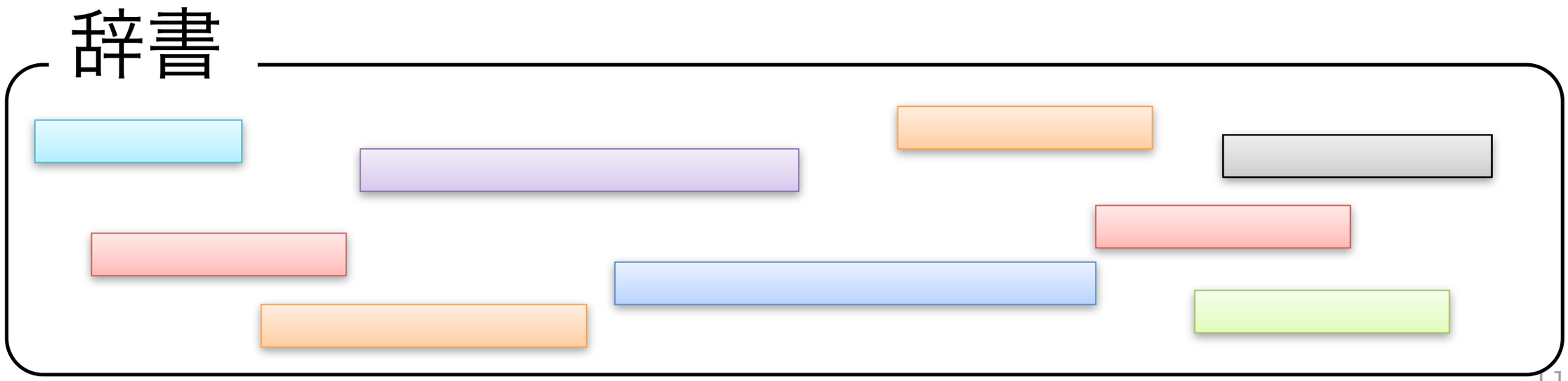
## 重なりがある部分文字列ベース

- 部分文字列は前の部分文字列の一部に影響される。
- P(ABCDEF) = P(AB|ε) * P(CDE|B) * P(F|DE)
  = P(AB) * P(BCDE) / P(B) * P(DEF) / P(DE)

# 確率の見積もり方

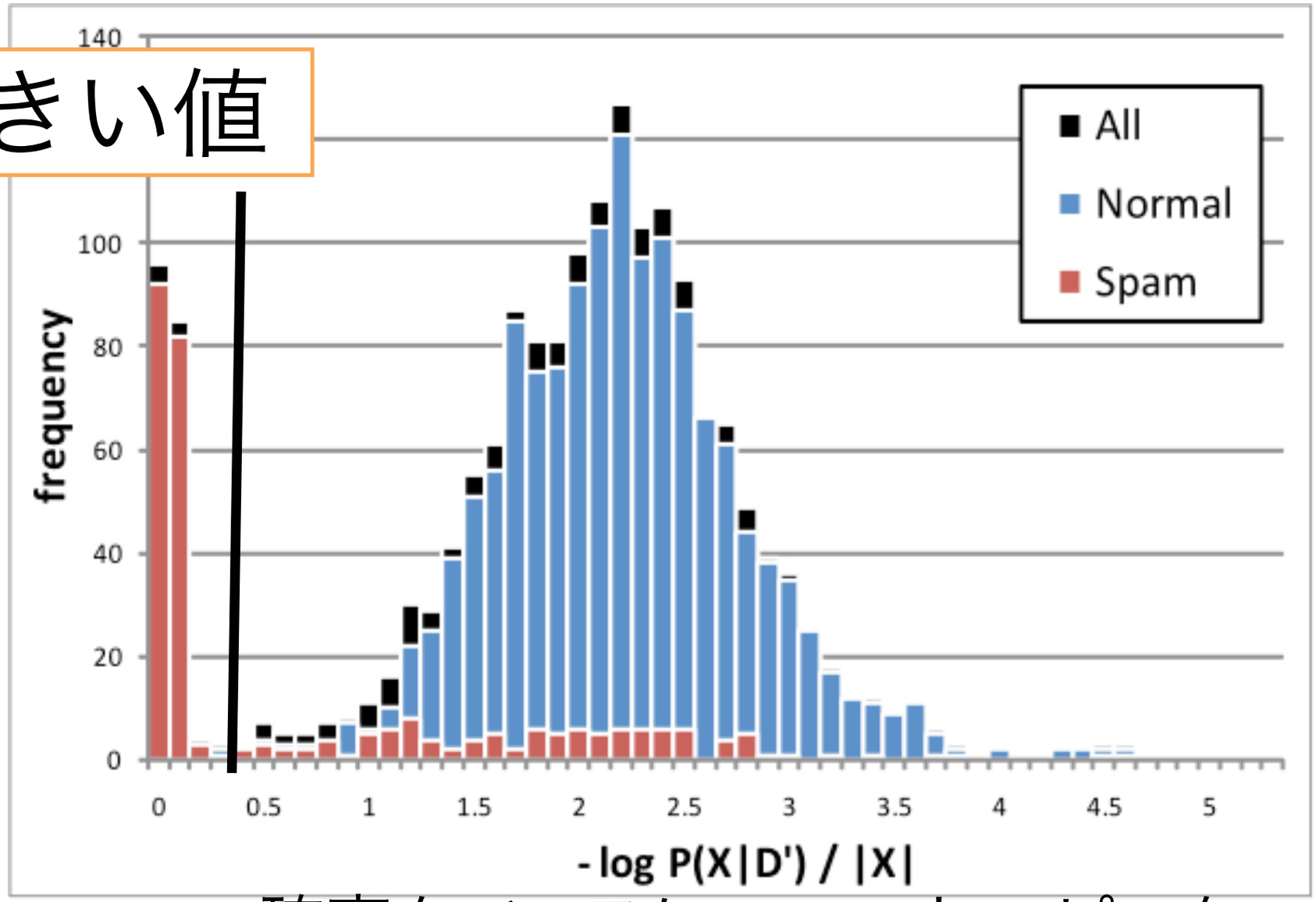
$$P(A\beta) = P(A) * P(B) / P(\alpha)$$




11

# 実験




確率をベースに、エントロピーを
長さで正規化したもの

# 意外な文字列パターン発見

Daisuke Ikeda and Einoshin Suzuki:
``Mining Peculiar Compositions of Frequent Substrings from Sparse Text Data Using Background Texts'',
ECML PKDD (LNAI 5781), pp. 596-611, Sep. 2009.

# Motivation

## 例外的な例、事象、データを見つける。

- データマイニングにおいて例外発見の有効性は示されている。
- 科学的な発見は従来の理論では説明できない例外的な事象から始まる。
- スパースなテキストでは、「頻度が低い」パターンは非常に多くある。

## 従来はデータを説明する手法が多い。

- 機械学習では与えられたデータを説明する規則を発見。
- 従来のデータマイニングでは頻出する (多くの事例にマッチする) パターンを発見。

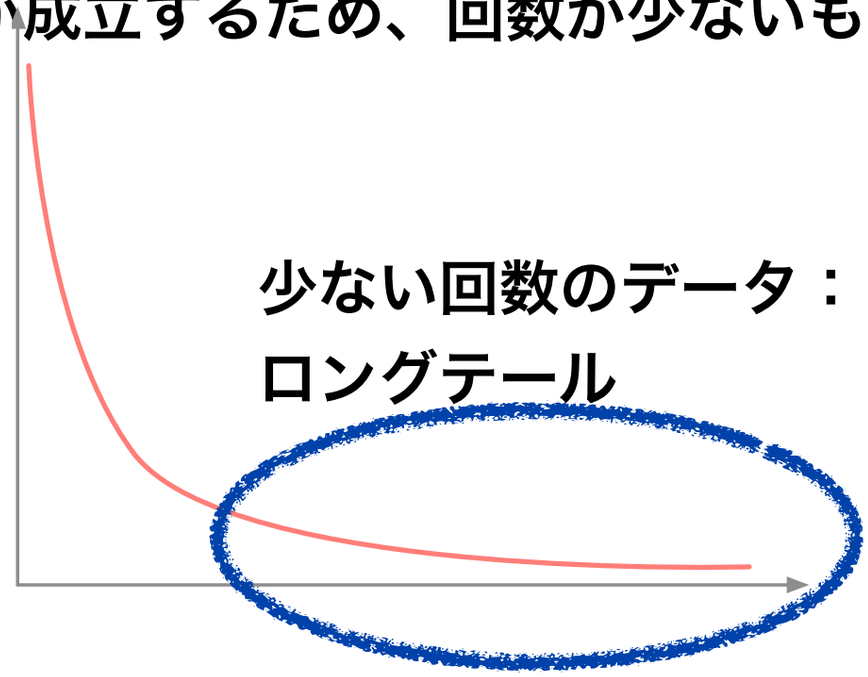
# Challenging

**「普通」を見つけてから、例外・意外を見つける必要アリ。**

- いままでの手法と同じようなことをして、さらに例外をみつける分、より難しい。

**例外的なパターンが説明するデータは少ないが、少ないデータが全て重要なわけではない。**

- テキストデータではデータスパースネスが成立するため、回数が少ないものは非常に種類が多い。



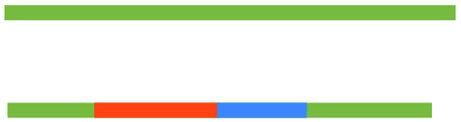
# 概要

## 入力データ

- 対象となるデータ以外に、普通を決めるための背景集合を用いる。

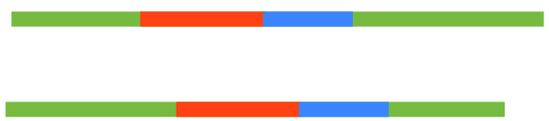
## 発見するパターン

- 頻出な語の連接。



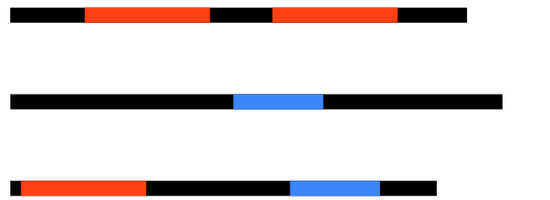
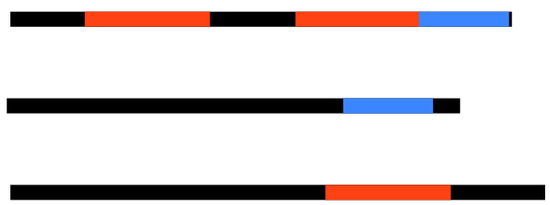
## 評価基準

- 背景集合との頻度の比



ターゲット集合

P(－), P(－)は小
P(－ －)は大

背景集合

P(－), P(－)が大
P(－ －)は小

# Motivating Example

Papers written by non-natives: Target Texts

.....we employ a set.....and.....

....in order to discuss about a vector space on a space.....
...such as the field of real or of complex numbers...

We are developing ..... algorithms .....about....

We discuss about a structure on t
.....namely we discuss a vect

.....is more frequent in.....and much more.....

discuss about:
意外な組み合わせ

Unlike existing methods..... use simple estimation ...
Simply discuss about a field...

⋮

Papers by native English speakers: Background Texts

.....we discuss various issues ....

Given ..... is to put about....., where I discuss.....and .....

..... about 10 oclock .....and....precisely ...about

⋮

discuss, about:

# 既存手法：z-score

**確率モデルを仮定し、これから推定される**
**文字列の出現頻度からのずれを指標とする**

$$z(w) = \frac{f(w) - E(w)}{N(w)}$$

$f(w)$ : observed frequency
$E(w)$ : expectation of frequency
$N(w)$ : normalization factor

## 部分文字列のζスコア

- **確率モデルとしてBernoulliモデルを仮定**
  **各文字の確率が既知で、文字ごとに独立に生起する**

$$\zeta(w) = \frac{f(w) - (n - |w| + 1)\hat{p}}{\sqrt{(n - |w| + 1)\hat{p}(1 - \hat{p})}}$$

# 確率モデルによる手法の問題点

**適切な確率モデルを選ぶ必要がある。**

- 複雑なモデルは学習コストが高い。

**頻度のずれを基準にしているので見つかったパターンの意味を捉えるのはむずかしい。**

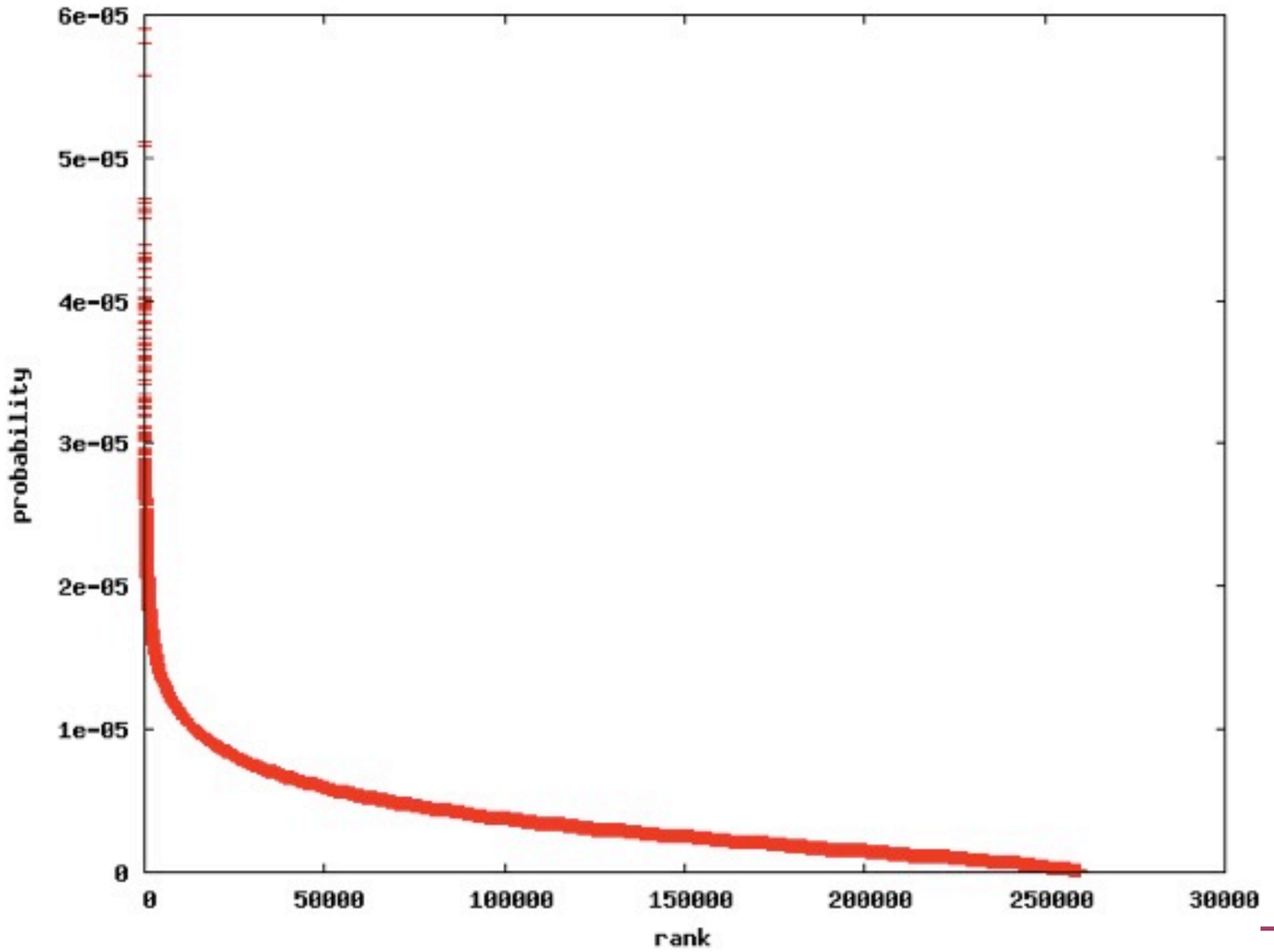
- 得られる情報は頻度(のずれ)。

**(テキストデータに多く成立する)スパースなデータの推定には向いていない。**

- Bernoulliモデルは各文字が独立。
- Markovモデルは重なりがあるが適切な次数が必要。
- どちらであっても、どの語(パターン)でも確率がふられる。

# データスパースネス

**与えられたデータ中には、出現しない**
**語(パターン)が数多くある**




出現確率

順位

# 意外な文字列パターン

$T, B$ : two sets of strings (ターゲット、背景)

$x, y$ : 文字列

$\theta > 1$ : しきい値

$x$ is contrastive w.r.t $\theta$ in target (resp. background)

$$\overset{\text{def}}{\iff} P(x|T) > \theta P(x|B) (\text{resp. } P(x|B) > \theta P(x|T))$$

$\theta_T, \theta_B > 1$: しきい値

$xy$ が(Tにおいて Bに対して)意外である

$$\overset{\text{def}}{\iff} \quad P(xy|T) > \theta_T P(xy|B)$$
$$P(x|B) > \theta_B P(x|T)$$
$$P(y|B) > \theta_B P(y|T)$$

# アルゴリズム

接尾辞木と呼ばれるデータ構造を用いて
全てのx, y, xyの候補に対しpeculiarかどうかをチェック

- ただしxyは$\eta_T$回以上出現しないといけないので、
  木の巡回中にあるノード(xyに対応)の出現回数が$\eta_T$回
  未満になれば、それより下のノードは巡回しない。

## $O(N^2)$のアルゴリズム

- ナイーブな方法では $O(N^4)$。
- パターンの形式がanti-monotonicityを満さないので、
  $O(N^2)$より早くするのは難しいと予想される。

# 実験：Tgt=枯草菌, Bkg=大腸菌

$(\theta_T, \theta_B, \eta_T) = (1.1, 5, 10)$

| $(x, y)$ | $\lvert xy \rvert$ | $(f(xy\lvert T), f(xy\lvert B))$ | $(f(x\lvert T), f(x\lvert B))$ | $(f(y\lvert T), f(y\lvert B))$ | $z$-score |
|---|---|---|---|---|---|
| (*CGGCGTGG, ACTACCAG*) | (8, 8) | (10, 7) | (66, 450) | (19, 154) | 3.57E+02 |
| (*CTGGTAGT, CCACGCCG*) | (8, 8) | (10, 7) | (19, 154) | (66, 450) | 3.57E+02 |
| (GCGTGG, ACTACCAG) | (6, 8) | (10, 7) | (529, 3845) | (19, 154) | 7.76E+01 |
| (*GGCGTGG, ACTACCAG*) | (6, 8) | (10, 7) | (161, 1407) | (19, 154) | 1.67E+02 |

$(\theta_T, \theta_B, \eta_T) = (10, 2, 15)$

| $(x, y)$ | $\lvert xy \rvert$ | $(f(xy\lvert T), f(xy\lvert B))$ | $(f(x\lvert T), f(x\lvert B))$ | $(f(y\lvert T), f(y\lvert B))$ | $z$-score |
|---|---|---|---|---|---|
| (*CAGCG,GCGCC*) | (5, 5) | (17, 0) | (9816, 24161) | (6759, 17014) | 8.924 |
| (*GGCGC, CGCTG*) | (5, 5) | (17, 0) | (6759, 17014) | (9816, 24161) | 8.924 |
| (*CGCG, GCGCC*) | (4, 5) | (16, 1) | (16950, 56436) | (6759, 17014) | 2.235 |
| (*GGCGC, CGCG*) | (5, 4) | (16, 1) | (6759, 17014) | (16950, 56436) | 2.235 |

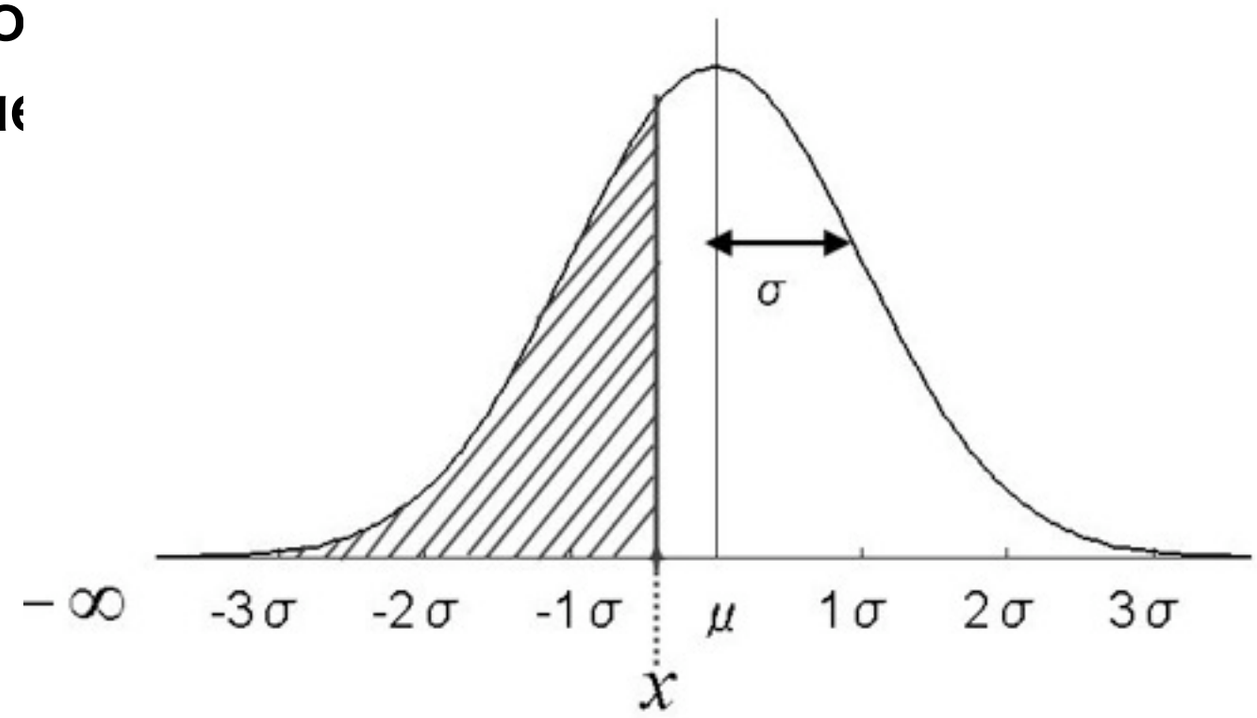
$z$-score is calculated with Bernoulli model, where

$$p(A) = p(T) = 0.282, p(C) = p(G) = 0.218$$

# Comparison with z-score

## z-score: standard normal distribution

- we can estimate the number o
  which have some z-score valu
- この入力ではz-scoreが
  2以上のものは100万程度
  存在する

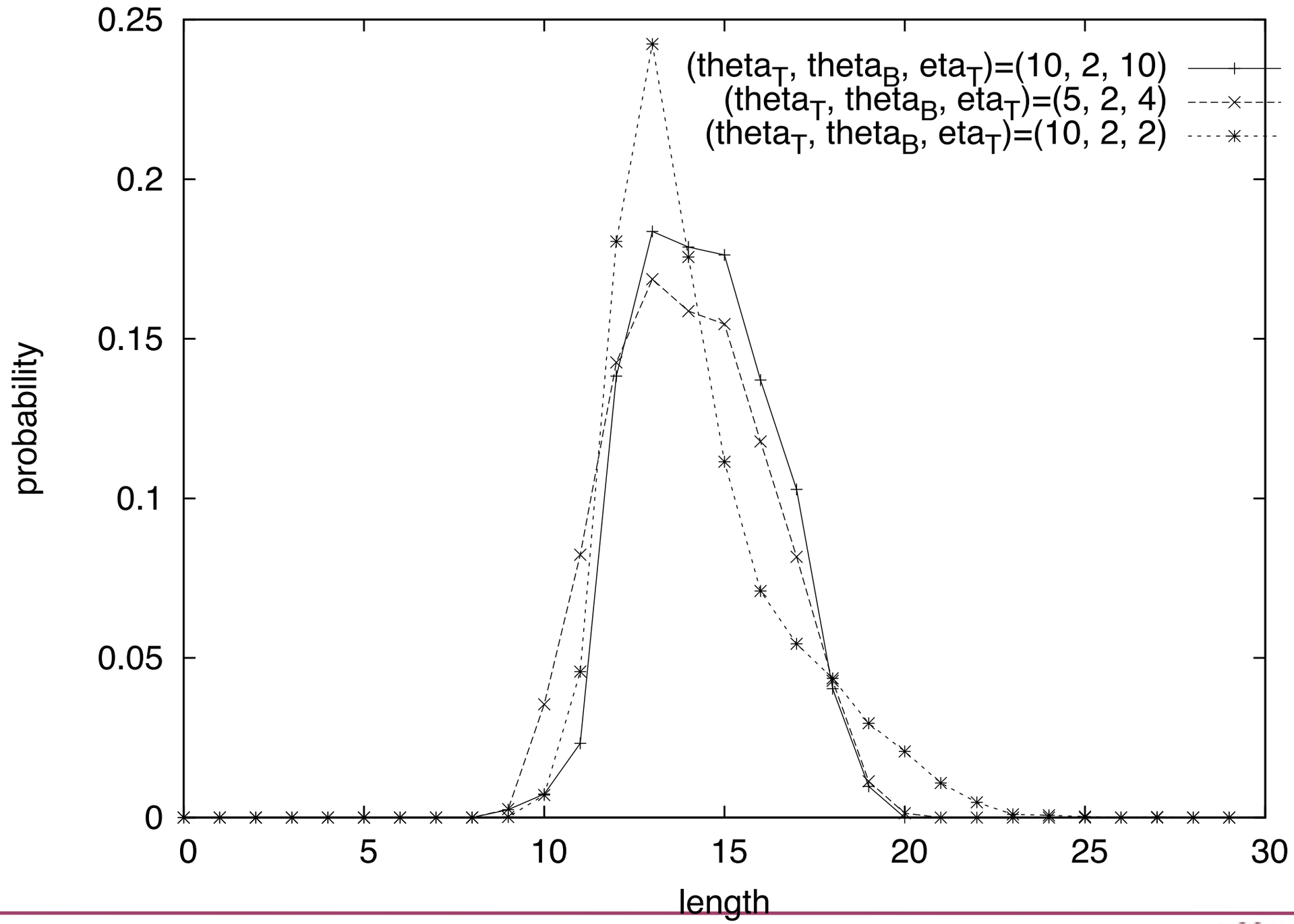



$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$E(X) = \mu$$

$$V(X) = \sigma^2$$

# Length of Peculiar Compositions

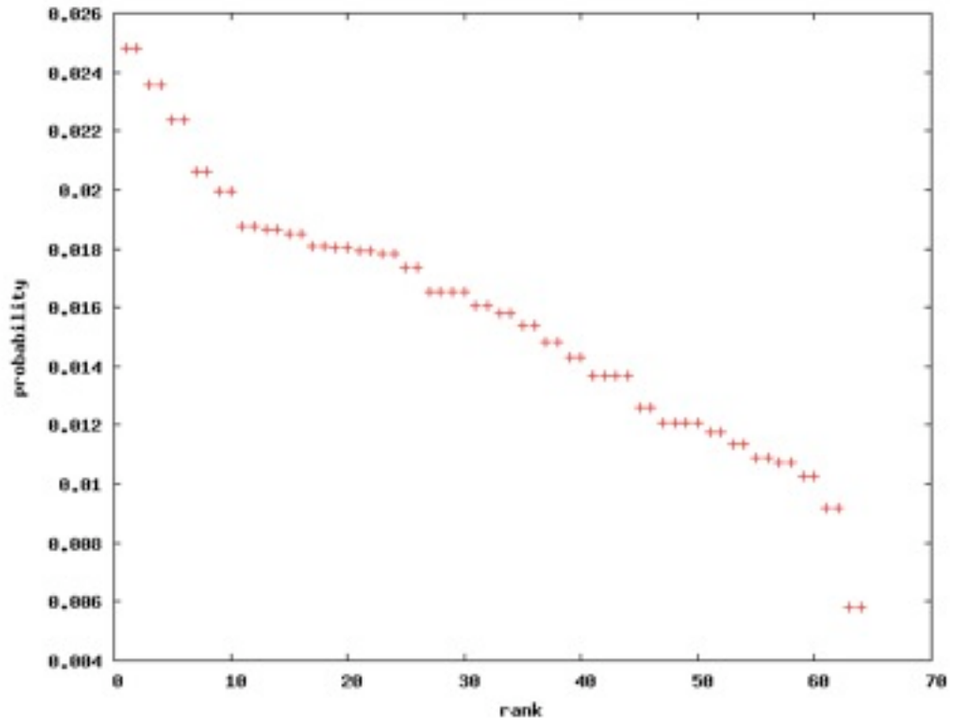


# 長さの分布：z-score

**z-scoreが10以上(または100以上)で、かつ、頻度が10以上のものを出力し、この時の見つかった文字列の長さを出力**

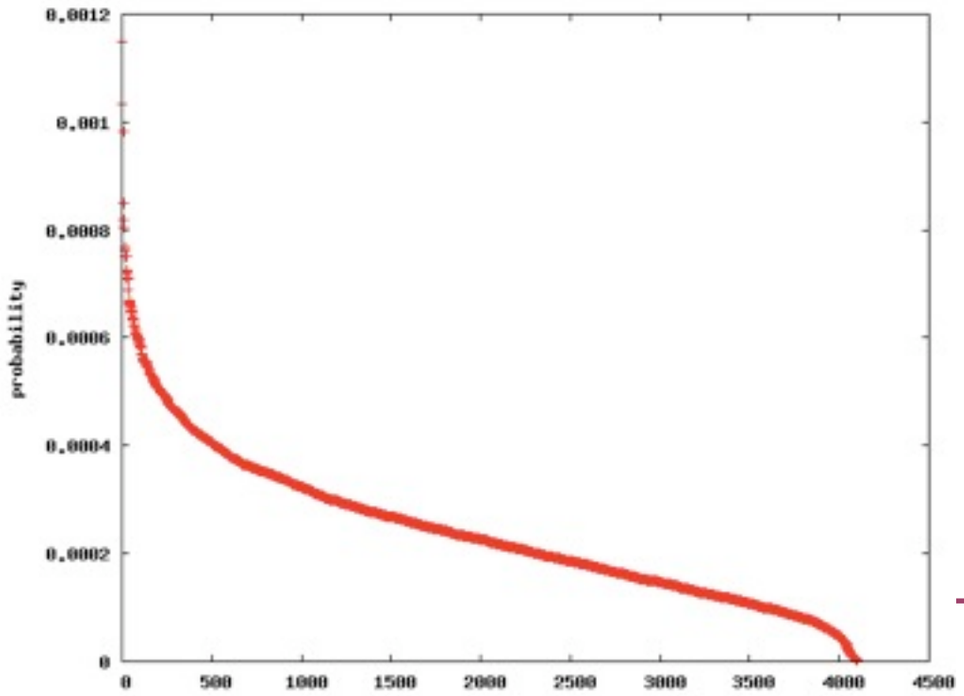
| 長さ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| >10 | 3 | 2 | 7 | 6 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 11 | 3 | 1 | 1 | 1 | 1 | 1 |
| >100 | 9 | 30 | 108 | 352 | 1090 | 3040 | 6555 | 9840 | 10543 | 8666 | 1974 | 55 | 4 | 4 | 2 | | | | | |

# 長さごとの文字列の頻度分布




n=3

n=6




n=9