形態素周辺確率を用いた 確率的単語分割コーパスの構築とその応用

岡野原 大輔† 工藤 拓‡ 森 信介§
†東京大学情報理工学系研究科コンピュータ科学専攻
‡Google Japan §日本 IBM 東京基礎研究所
hillbig@is.s.u-tokyo.ac.jp, taku@google.com, mori@fw.ipsj.or.jp

1 概要

本稿では、入力テキストの基本単位への分割情報を曖昧性を保ったままコンパクトに保持し、オンライン時に適切な処理単位を求める手法を提案する．情報検索や文書分類などにおいては、与えられたテキストを単語など適切な処理単位に分割した上で扱う場合が多い．この時、どのような分割が最適かはタスク依存であり決定できないため、曖昧性を保ったまま分割情報を保持することが望まれる．本稿では CRF を用いた形態素解析結果の周辺確率を用いて各文字間の分割確率を求めた上でそれを保持し、任意部分文字列の出現確率（確信度）をオンラインで効率良く計算する方法を提案する．また提案手法と従来手法を用いた全文検索の実験結果を示し、提案手法が各クエリに対し、より適切な出現確率を与えられることを示す．

2 はじめに

情報検索や文書分類において、入力テキストをどのように表現するかは古くから存在する問題である．日本語等の分かち書きされていない言語の場合，多くは入力されたテキストを形態素解析によって形態素（単語）に分解した上で，その単語列に対して処理を行う．例えば情報検索では各形態素単位での出現位置を記録した索引を構築し，文書分類では形態素の頻度付き集合（bag-of-words; BOW) によって文書を表現し，それを分類に利用する．しかし，どのような分割が最適であるかはタスク依存であり決定できないため，分割を一意に決定するのは難しく結果として応用アプリケーションの性能低下につながる．例えば，形態素解析の結果によっては，単語が実際に出現しているのにも関わらず存在しない場合（再現率の低下）や，関係の無い単語が存在していると判断される（精度の低下）場合が起こりうる．この問題は形態素解析の精度が向上すれば解決できる問題ではない．例えば，以下に示す複合語で構成された例は，どの分割が最適かを一意に決定するのは難しい問題である [6].

- 本部長 本部/長 or 本/部長
- 応援団員 応援団/員 or 応援/団員
- 横浜市役所 横浜/市/役所 or 横浜市/役所 or 横浜/市役所

そのため，形態素解析の結果で最適な単語分割列を一つ決定し、それのみを後の処理で使うのではなく，複数の解析結果を保存しておくことが考えられる．例えば，全ての部分列について，単語として尤もらしいかの確率情報を保存することが考えられる．しかし，この方法は，確率値が高いものや単語長が一定以下であるもののみを残す制限を加えたとしても，作業領域量は非常に大きく，実用上の大きな支障となる．

本稿では，この確率情報を各文字間の分割確率という factorize された状態で保持することで，テキスト長に比例する程度のコンパクトなサイズで保持し，これを利用して任意部分列がその文脈に出現した確率をオンラインで効率良く求める方法を提案する．各文字間の分割確率は CRF による形態素解析結果の周辺確率として求められ，また，各文字間の分割確率がクエリに対しても与えられている場合においてもクエリの出現確率を効率良く求めることができることを示す．Wikipedia に対する全文検索タスクにおいて本手法を適用した結果は従来結果に比べ精度が高く，適切な確率情報を与えられていることを示す．

3 背景

3.1 CRF: 条件付確率場

Conditional Random Fields (CRF) [1] は，1 つの指数分布モデル（最大エントロピーモデル）によって各出

力系列 $\mathbf{y}$ の入力列 $\mathbf{x}$ に対する条件付き確率 $P(\mathbf{y}|\mathbf{x})$ を表現する．

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{i=1...|\mathbf{y}|} \sum_{k} \lambda_k f_k(\mathbf{x}, y_{i-1}, y_i, i)\right)$$

ただし，$Z(\mathbf{x})$ は正規化項であり，

$$Z(\mathbf{x}) = \sum_{\mathbf{y'}} \exp\left(\sum_{i=1...|\mathbf{y}|} \sum_{k} \lambda_k f_k(\mathbf{x}, y'_{i-1}, y'_i, i)\right)$$

が成り立つ．形態素解析の場合は $\mathbf{x}$ が入力文字文に，$\mathbf{y}$ が形態素解析結果に相当する．特に本稿で扱うモデルでは，各 y_i は，単語の開始位置であることを示す B と単語中の位置であることを示す I の二種類からなる．

CRF は識別モデルであるため HMM をはじめとする生成モデルでは扱うことが難しい重複する特徴を柔軟に取り込むことが可能である．例えば形態素解析においては，階層構造を持つ品詞体系，文字種や文字列といった特徴が重要であるが，これらの特徴を自由に設計しモデルに組み込むことが可能となる．入力 $\mathbf{x}$ に対する最適な出力 $\mathbf{y}^*$ は，Veterbi アルゴリズムを用いて効率的に求められ，また後で述べる $y_i = B$ である事象の周辺確率も forward-backword アルゴリズムと同様の方法で効率的に求められる．

CRF は，学習データに対する最尤推定を用いてパラメータを推定する．この時パラメータ数が学習データ数に比べて大きい時，過学習を引き起こす．この過学習を防ぐためにパラメータの正則化を行う．これは事後確率最大化 (MAP) 推定とも呼ばれる．CRF の詳細については，[1] を参照．CRF を用いた形態素解析については [5] を参照．

3.2 SSC: 確率的単語分割コーパス

SSC (確率的単語分割コーパス，Stochastically Segmented Corpus)[3] は，生コーパス $x_{1...n}$ とその連続する各 2 文字 x_i, x_{i+1} の間に単語境界が存在する確率 P_i の組として定義される．x_1 の前と x_n の後に単語境界が存在するとみなすことにする．確率変数 X_i を，

$$X_i = \begin{cases} 1 & x_i,\, x_{i+1} \text{ の間に単語境界が存在する場合} \\ 0 & x_i,\, x_{i+1} \text{ が同じ単語に属する場合} \end{cases}$$

とし（$P(X_i = 1) = P_i, P(X_i = 0) = 1 - P_i$），各 $X_0, X_1, ..., X_{n-1}$ は独立であることを仮定する．SSC を保存するのに必要な作業領域量は $O(n)$bit であり，P_i を $k$bit で表現した場合に必要な作業領域量は元の文書に加え knbit が必要である．

SSC においてクエリ $q[0...m-1]$ が文章中の位置 $x[t...t+m-1]$ に出現している必要十分条件は，次の 4 つである [3].

1. 文字列が等しい ($x[t...t+m-1] = q[0...m-1]$)
2. $x[t]$ の直前に単語境界がある ($X_t = 1$)
3. 単語境界が文字列中に無い ($X_j = 0, t+1 \leq \forall j \leq m-1$)
4. $x[t+m-1]$ の直後に単語境界がある（$X_{t+m} = 1$）

この条件より，$x[t...t+m-1]$ が単語 $q[0...m-1]$ である確率 p は $q[0...m-1] \neq x[t...t+m-1]$ の時は 0，それ以外の時は次の通りに求められる [3].

$$p = P_t[\Pi_{j=t+1}^{t+m-1}(1 - P_j)]P_{t+m}$$

図 1 に各手法により確率的単語分割コーパスの各文字間の分割確率を求めた例を示す．

3.3 全文検索

全てのテキスト中に出現した文字列に対する検索を全文検索と呼ぶ．これに対応し，単語や部分列がテキスト中に出現している位置を前もって求めておくことにより高速な検索を可能とする索引の中で，全文検索を実現するものを全文索引と呼ぶ．全文索引の中で特に任意の部分列を検索可能な文字索引[1]である n-gram 方式や Suffix Arrays は，従来の転置ファイルなどと比べ作業領域量が大きい問題が知られていたが，近年では作業領域量の小さいものも提案されている [4].

全文検索では，得られた結果のうち，クエリと実際に一致しているかが問題となる．例えば，“京都” を検索した場合，全文検索では，“東京都” が出現している箇所も文字上は一致するので検索結果として報告するが，この二つは違うものであり検索結果としては適さない．このため，形態素解析の情報を検索結果に組み込む必要がある．提案手法では，形態素解析結果を SSC の形で，あらかじめ索引構築時に一緒に求めておき，それを利用することで，全文検索の結果得られた全ての検索結果候補に対しての出現確率（クエリとの一致の確信度）を求めることが可能となる．

4 CRF からの SSC の構築

確率的単語分割コーパスを構築するにあたって，既存研究では全ての単語境界の確率推定に同一パラメータ

[1] これに対し形態素解析の結果等を用いてテキストを単語に区切っておき、単語の出現位置だけを記録したものを単語索引と呼ぶ.

	確	率	的	単	語	分	割	コ	ー	パ	ス
形態素解析結果	1	0	1	1	0	1	0	1	0	0	0	1
SSC (α=0.95)	0.95	0.05	0.95	0.95	0.05	0.95	0.05	0.95	0.05	0.05	0.05	0.95
SSC (CRF)	0.99	0.01	0.99	0.89	0.18	0.85	0.19	0.95	0.0	0.0	0.0	0.99

“単語分割”の出現確率

形態素解析結果 0.00 = 1 * (1-0) * (1-1) * (1-0) * 1
SSC(α=0.95) 0.04 = 0.95 * (1-0.05) * (1-0.95) * (1-0.05) * 0.95
SSC(CRF) 0.08 = 0.89 * (1-0.18) * (1-0.85) * (1-0.19) * 0.95

図 1: “確率的単語コーパス” を，形態素解析結果の最適な結果を用いて分割した場合，SSC のパラメータが $\alpha = 0.05$ の場合（形態素解析結果の 1 を $1-\alpha$，0 を α にした場合，CRF の形態素周辺確率を用いて推定した場合を示す．“単語分割” が出現した確率の結果も示す．

α を用いて，形態素解析の結果単語境界であると判断された境界については $P_i = \alpha$, そうでない点については $P_i = 1-\alpha$ としていた [2, 3]．この場合，文脈に依存した形で単語境界の確率推定が行えない問題がある．また，CRF を用いた場合には確率情報が付与されるが，最適な出力を用いた場合はこの確率情報は利用されない．例えば，解析結果として，非常に確率が近い二通りの分割の仕方があったとしても，片方だけが採用され，もう一方の有用な情報は捨てられてしまう．本稿では CRF 上の全系列の情報を周辺確率として利用するため上記の問題を解決できる．本稿では B, I タグ付け問題として単語分割問題を扱う．$y_i = B$ は $X_i = 1$ を表し，$y_i = I$ は $X_i = 0$ を表す．

CRF では，各文字間の分割確率 P_i を分割されている事象の周辺確率を用いて求める．これは以下のようにして求められる．

$$\begin{aligned} P_i &= \sum_{y_1,...,y_{i-1},y_{i+1},...,y_n} p(y_1,...,y_{i-1},B,y_i,...,y_n|\mathbf{x}) \\ &= \frac{1}{Z(\mathbf{x})}\frac{\alpha_{i,B}\beta_{i,B}}{e_{i,B}} \end{aligned} \quad (1)$$

ただし，$e_{i,y}$ は一つのラベルにしか依存しない emission コストであり，α_i と β_i で重複して数えているため除かれる．

$$e_{i,y} = \exp\left(\sum_k \lambda_k f_k(y,x,i)\right) \quad (2)$$

また,α_i，β_i は forward-backward コストであり，以下の再帰的な定義で与えられ動的計画法で求められる．

$$\begin{aligned} \alpha_{i,y} &= \sum_{y'}\left(\alpha_{i-1,y'}\cdot\exp(\sum_k \lambda_k f_k(y',y,x,i))\right) \\ \beta_{i,y} &= \sum_{y'}\left(\beta_{i+1,y'}\cdot\exp(\sum_k \lambda_k f_k(y,y',x,i))\right) \end{aligned}$$

5 クエリの出現確率の効率的な計算

各単語間に境界があるかどうかは独立事象として扱っているため，ある部分列が単語である確率は図 2 の *calcProb* のように同時確率として効率的に計算される．

次にクエリもテキストと同様に，確率的分割が行われている場合を考える．この時，クエリの全ての分割を考慮したスコアは，q の全ての分割方法は $O(2^{|q|})$ 通りあるので，そのままこれを求めると計算量が大きいが，動的計画法を用いることで，図 2 の *calcProbQ* のようにクエリ長に比例する計算量で効率的に求められる．各文字間において, そこで分割が起こった確率 (図 2 *prevB*）と起きなかった確率（図 2 *prevI*) を求め，これを次の分割時に利用する．

6 実験

本手法の有効性を調べるために，Wikipedia 日本語版[2]を利用した全文検索の結果を調べた．CRF の学習には CRF++[3]を用いた．検索対象テキストに対し CRF を用いて形態素解析を行い, その結果を用いて, 各文字間の

[2] http://download-yaseo.wikimedia.org/jawiki/
[3] http://chasen.org/ taku/software/CRF++/

```
function calcProb (t, len)
# t:  検索対象テキスト中のマッチした位置
# len:  クエリ長
    prob = p[t]
    for i=1 to len-1
        prob = prob * (1- p[t+i])
    return prob * p[t+len]
end

function calcProbQ (t, len, q[])
# q[]:  クエリに対する確率的単語分割情報
    prevB = p[t]
    prevI = 0
    for i=1 to len-1
        tmp = prevB + prevI
        prevB = tmp * q[i] * p[t+i]
        prevI = tmp * (1-q[i]) * (1-p[t+i])
    end
    return (prevB + prevI) * p[t+len]
end
```

図 2: *calcProb* は確率的単語分割コーパス中の $x[t...t+m-1]$ が単語である確率を返し,*calcProbQ* は, クエリも確率的単語分割コーパスで表現されている場合に, クエリが $x[t...t+m-1]$ に出現した確率を返す．$p[i]$ は対象テキスト中の i と $i+1$ 文字目の間が分割されている確率であり,$q[i]$ はクエリ中の i と $i+1$ 文字目が分割されている確率である．

境界確率を (1) に基づいて求めた．クエリに対する確率的分割操作は行っていない（クエリの両端の分割確率が 1, それ以外は 0）．被験者二人に対し，各クエリの結果を，見つかった順に順位をつけた場合（ベースライン）と，提案手法による確率付けで確率の高い順に順位をつけた場合（提案手法）の上位 20 件につき，それぞれクエリが指している対象と一致しているかどうかを合っているかどうかの二択で判断してもらった．利用したクエリは次の１０件 “京都”，“トロ”，“本部”，“国際”，“スター”，“パン”，“１月”，“かしい”，“はかた”，“国” である．表 3 に．1 位，5 位，10 位，20 位までのそれぞれの検索結果とその精度を示す．精度は被験者が一致していると判断した個数/候補数として計算される．この結果から，提案手法が安定して正しい検索候補を高い順位, つまり高い確率で選んでいることが分かる．

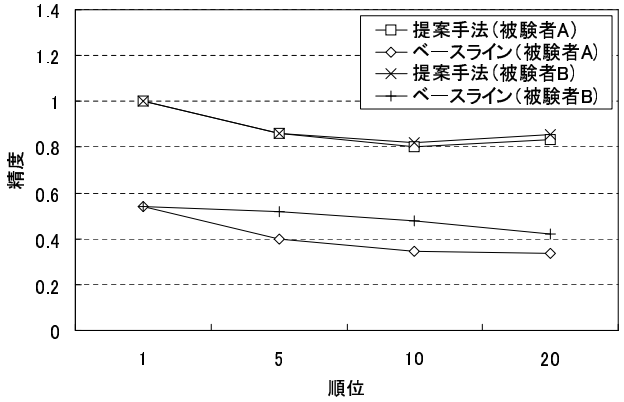

図 3: 被験者 A,B に対しクエリ 10 件の検索結果の上位 20 件のそれぞれについてクエリが指している対象と一致しているかどうかを判断してもらいその精度を比較した結果．提案手法：本論文で提案した方法により各出現場所がその単語の通り区切られている確率によってソートした結果．ベースライン：見つかった順に表示した結果．

7 まとめ

本論文では，入力テキストの基本単位への分割情報を各文字間の分割確率という形で曖昧性を保ったままコンパクトに保持し，任意の部分列の出現確率を効率的に求める方法を提案した．提案手法は CRF の結果を利用しているため，高精度であり，未知語などに対しても適切な確率を与えられる．また，提案手法とベースラインを全文検索タスクにおいて比較し，提案手法を用いることにより適切なスコアを与えられることを示した．提案手法はクエリが確率的分割されている場合でも効率的な処理が可能である．また，本提案手法は情報検索のみならず，文書分類や情報抽出などにも同様に適用されることが期待される他，係り受け関係など，より複雑な情報についても factorize された形で確率情報を保持することで精度，計算量，作業領域量のバランスがとれた処理が可能になると考えられる．

参考文献

[1] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML 2001*, 2001.

[2] Shinsuke Mori and Daisuke Takuma. Word n-gram probability estimation from a Japanese raw corpus. In *Proc. of ICSLP 2004*, 2004.

[3] Shinsuke Mori, Daisuke Takuma, and Gakuto Kurata. Phoneme-to-text transcription system with an infinite vocabulary. In *Proceedings of the 21st International*

Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 729–736, Sydney, Australia, July 2006. Association for Computational Linguistics.

[4] G. Navarro and V. Makinen. Compressed full text indexes. Technical report, Dept. of Computer Science, University of Chile, 2006.

[5] 工藤 拓. Conditional random fields を用いた日本語形態素解析. In **情報処理学会自然言語処理研究会** *SIGNL-161*, 2004.

[6] 工藤 拓. 形態素周辺確率を用いた分かち書きの一般化とその応用. In **言語処理学会全国大会** *NLP-2005*, 2005.