

# 優先度学習を用いた文短縮手法

牧野 恵<sup>†</sup> 平尾 努<sup>††</sup> 山本 和英<sup>†</sup> 磯崎 秀樹<sup>††</sup>

<sup>†</sup> 長岡技術科学大学 電気系 <sup>††</sup> 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所  
E-mail: <sup>†</sup>{makino,ykaz}@nlp.nagaokaut.ac.jp <sup>††</sup>{hirao,isozaki}@cslab.kecl.ntt.co.jp

## 1 はじめに

近年、ニュースの文字放送や字幕を自動的に作成する技術への需要が大きくなっている。これらは人間の読むスピードや表示する機器の大きさ等の制限により、ニュースをより圧縮したものでなくてはならない。伝統的な重要文抽出による要約では、指定された文字数内により多くの情報を含めることが難しい。そこでより高圧縮な要約を目指し、文短縮技術の研究が盛んに行われている [1, 2]。

例えば、堀ら [2] は文短縮を単語重要度を最大に且つ、日本語としてできるだけ自然な部分単語列を抽出する組合せ最適化問題として定式化し、動的計画法を用いて解く手法を提案している。一般的に、こうして得た最適解は常に最良であるとは限らず、他にもより良い解が存在する可能性がある。そこで本稿では堀らの手法に対し、優先度学習を適用した高性能な文短縮手法を提案する。評価実験では10位までの短縮文ランキングを優先度学習を用いて再ランキングしたことにより、1位の BLEU スコアを約3ポイント改善できた。また BLEU スコアが最大となる順位平均も優先度学習後では約2位改善でき、提案手法の有用性を確認した。

## 2 準備

堀らによると文短縮は、 $N$  個からなる入力文の単語列から  $M$  個の単語列を抽出する問題として定式化できる。いま10個の単語から成る入力文に対し、出力文(短縮文)として5個の単語を抽出する問題を図1を用いて説明する。

横軸  $i$  は出力文(短縮文) $V$  の単語インデックスを表し、

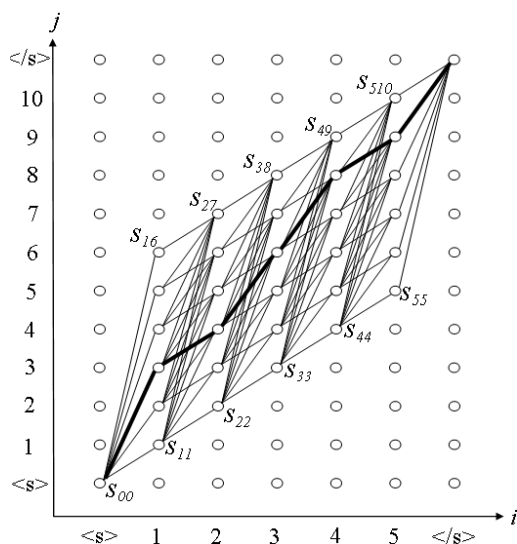


図1 動的計画法の探索領域

縦軸  $j$  は入力文  $W$  の単語インデックスを表す。なお入力文  $W$  の前後にはあらかじめ文頭記号  $\langle s \rangle$ 、文末記号  $\langle /s \rangle$  を挿入することにより初期状態と最終状態を明確にする。 $s_{ij}$  はステート(単語)を表す。各ステートの重み  $W(s_{ij})$  は単語の重要度を表し、各ステート間(リンク)の重みは単語接続の良さを表す。重要な単語を含み且つ、日本語としてできるだけ自然な部分単語列を抽出するには、ステートの重みとリンクの重みの和を最大にするようなパスを求める問題に帰着できる。さらに図1は文頭から文末に向い、抽出し得る部分単語列の全ての組合せを2次元空間に示したものであり、探索領域が図中の平行四辺形内のパスに限られることから、動的計画法で解くことができる。よってこの問題はステート列  $S = s_{00}, s_{1j'}, \dots, s_{M+1j'}$  のスコア  $Score(S)$  を最大にするステート列  $\hat{S}$  を動的計画法を用いて求める問題となる。なおこのとき  $j'$  は直前のステートから接続可能なステートの列の入力文の単語インデックス ( $i \leq j' \leq N - M + i$ ) を表す。

$$\hat{S} = \arg \max_S Score(S) \quad (1)$$

ここで  $Score(S)$  は次式で表される。

$$Score(S) = \alpha \sum_{i=1}^M W(s_{ij}) + \beta \sum_{i=0}^M W(s_{ij}|s_{i+1j'}) \quad (2)$$
$$(i+1 \leq j' \leq N - M + i + 1)$$

但し  $W(s_{ij})$  は  $tf \cdot idf$  を用いたステート  $s_{ij}$  のステートの重みを表し、 $W(s_{ij}|s_{i+1j'})$  は単語バイグラム確率<sup>\*1</sup>と係り受け確率を用いたリンクの重みを表す。また  $\alpha, \beta$  はステートの重みとリンクの重みに対して与えるパラメータである。例えば図1においてスコアを最大にするステート列が「 $s_{00}, s_{13}, s_{24}, s_{36}, s_{48}, s_{59}, s_{611}$ 」<sup>\*2</sup>であったとする。このステート列を単語に置き換えると出力される短縮文は入力文の「 $\langle s \rangle, w_3, w_4, w_6, w_8, w_9, \langle /s \rangle$ 」となる。図2に動的計画法を用いた前向き探索のアルゴリズムを示す。 $link_f(s_{ij})$  はステート  $s_{ij}$  と接続可能なステートの集合である。例えば図1で  $link_f(s_{13})$  とは

$$link_f(s_{13}) = \{s_{24}, s_{25}, s_{26}, s_{27}\} \quad (3)$$

となる。なお各ステートでは初期状態からそのステートまでの最適経路の最大スコアを保持する(5, 6行目)。

## 3 提案手法

上述の方法で得た短縮文が常に最良であるとは限らないため、本稿では優先度学習を適用することで、より良い短縮文を得る手法を提案する。優先度学習を行うためにはい

<sup>\*1</sup> 堀らは単語トライグラム確率を用いているがここでは簡略化し単語バイグラム確率を用いて説明を行う。

<sup>\*2</sup> 図1の太線。

```

1 for (i = 0; i ≤ M + 1; i++) do
2   for (j = i; j ≤ N - M + i; j++) do
3     for si+1j' ∈ linkf(sij) do
4       φ'(si+1j') ← φ(sij) + αW(si+1j')
                    + βW(sij|si+1j')
5       if φ'(si+1j') > φ(si+1j') then
6         φ(si+1j') ← φ'(si+1j')
7       endif
8     end
9   end
10 end

```

図2 動的計画法を用いた前向き探索

```

1 open ← {sM+1N+1}
2 closed ← φ
3 g(sM+1N+1) ← 0
4 f(sM+1N+1) ← h(sM+1N+1)
7 while (open ≠ φ) do
8   sij ← arg maxsij ∈ open f(S)
9   if (i = 0) then return success
10  open ← open - {sij}
11  closed ← closed ∪ {sij}
12  for si-1j' ∈ linkb(sij) do
13    g'(si-1j') ← W(si-1j'|sij) + W(sij) + g(sij)
14    f'(si-1j', si-1j') ← g'(si-1j') + h(si-1j')
15    if f(si-1j') < f'(sij, si-1j') then
16      g(si-1j') ← g'(sij)
17      f(si-1j') ← f'(sij, si-1j')
18      open ← q(si-1j')
19      q(si-1j') ← sij
20      if si-1j' ∉ open then
21        open ← open ∪ {si-1j'}
22      endif
23    endif
24  end
25 end
26 return open = φ

```

図3 後向き A\* アルゴリズム

くつかの短縮文候補が必要であるので、本稿では、後向き A\* を堀らの手法に適用することで効率的に N-best 解を得る。なお本稿では単語接続の良さを表すリンクの重みに単語バイグラム確率<sup>\*3</sup>と工藤ら [3] によって提案された係り受け確率を用いる。以下に N-best 解の導出法と優先度学習を説明する。

### 3.1 N-best 解の導出

堀らの手法は前向き DP であり、永田ら [4] が提案した後向き A\* を用いることで効率的に N-best 解を得ることができる。後向き A\* のアルゴリズムを図3に示す。

本稿の場合、永田らのアルゴリズムをそのまま用いると状態のスコアを誤って上書きしてしまうことがあるため、正確な探索ができるよう改良を行った。後向き A\* では初期状態 (s<sub>M+1N+1</sub>) から任意の状態 s<sub>ij</sub> までの最適経路のスコアを g(s<sub>ij</sub>) とし、最終状態 (s<sub>00</sub>) から s<sub>ij</sub> までの最適経路のスコアを h(s<sub>ij</sub>) とすると、s<sub>ij</sub> を通る最適経路のスコア f(s<sub>ij</sub>) は次式で与えられる。

$$f(s_{ij}) = g(s_{ij}) + h(s_{ij}) \quad (4)$$

<sup>\*3</sup> 堀らは単語接続の良さを表すリンク重みに単語トライグラム確率を用いていたが、本稿では計算効率を考え、単語バイグラム確率を用いた。

h(s<sub>ij</sub>) は前向き DP によって与えられた最適経路のスコアを用いる。また g(s<sub>ij</sub>) は次式で表される。

$$g(s_{ij}) = W(s_{ij}|s_{i+1j'}) + W(s_{i+1j'}) + g(s_{i+1j'}) \quad (5)$$

$$(i + 1 \leq j' \leq M - N + i + 1)$$

ここで W(s<sub>ij</sub>|s<sub>i+1j'</sub>) は単語バイグラム確率と係り受け確率を用いたリンクの重みであり、W(s<sub>i+1j'</sub>) は状態 s<sub>i+1j'</sub> の単語重要度を用いた状態の重みである。また g(s<sub>i+1j'</sub>) は初期状態 (s<sub>M+1N+1</sub>) から状態 s<sub>i+1j'</sub> までの最適経路のスコアを表す。後向き A\* では open と closed というリストを用いる。open リストには展開によって得られた新しい状態の集合を格納し、closed リストにはすでに展開された状態を格納する。link<sub>b</sub>(s<sub>ij</sub>) は状態 s<sub>ij</sub> が接続可能な状態の集合である。例えば図1において link<sub>b</sub>(s<sub>46</sub>) とは

$$link_b(s_{46}) = \{s_{33}, s_{34}, s_{35}\} \quad (6)$$

を表す。また q(s<sub>ij</sub>) は状態 s<sub>ij</sub> が指す直前の状態を表す。スコアの上書き部分では状態 s<sub>ij</sub> の直前の状態 q(s<sub>i-1j'</sub>) を open リストに加える。これにより次に状態 s<sub>i-1j'</sub> が選ばれる際にはその前にスコアの大きい状態 q(s<sub>i-1j'</sub>) が選ばれ、正しい経路を探索することができる (18 行目)。

## 3.2 優先度学習

### 3.2.1 Ranking Voted Perceptron

本稿では Collins ら [5] によって提案された Ranking Voted Perceptron を用いる。Ranking Voted Perceptron はオンライン型の優先度学習手法である。訓練データには k 番目の短縮課題で出力したシステムの N-best 解集合 S<sub>k</sub> とその短縮課題で与えられているリファレンス集合 T<sub>k</sub> から構成されたセット <S<sub>k</sub>, T<sub>k</sub>> を用いる。但し 1 つの短縮課題に対しリファレンスが複数存在するため、実際には k 番目の短縮課題に対して |T<sub>k</sub>| の数だけ訓練データが存在する。本稿では訓練データのセットを列挙し、以下のように表す。

$$\mathcal{C}(t_i) = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{iN+1}\} \quad (7)$$

$$t_i \in T_k$$

ここで x<sub>i1</sub> はリファレンスの素性ベクトルを表し、x<sub>i2</sub>, ..., x<sub>iN+1</sub> はシステムが出力した N 個の短縮文の素性ベクトルを表す。いまある素性ベクトル x が与えられたとすると、その優先度 G(x) と予測  $\hat{j}$  は次式で与えられる。

$$G(\mathbf{x}) = \sum_{ij} \alpha_{ij} (\mathbf{h}(\mathbf{x}_{i1}) \cdot \mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{x}_{ij}) \cdot \mathbf{h}(\mathbf{x})) \quad (8)$$

$$\hat{j} = \arg \max_{j=1 \dots N+1} G(\mathbf{x}_{ij}) \quad (9)$$

本稿での優先度学習はリファレンスが常に優先されるようにするため、訓練において予測  $\hat{j}$  が誤った場合には重み α<sub>ij</sub> の更新を行う。つまり  $\hat{j} \neq 1$  となった場合、リファレンスよりもシステム出力が優先されるため誤りとみなし重みの更新を行う。図4に Ranking Voted Perceptron の訓練アルゴリズムを示す。

### 3.2.2 素性ベクトル間の類似度

式(8)での  $\mathbf{h}(\mathbf{x}_{ij}) \cdot \mathbf{h}(\mathbf{x})$  は高次元空間における素性ベクトルの内積である。つまり  $\mathbf{h}(\mathbf{x}_{ij}) \cdot \mathbf{h}(\mathbf{x})$  は x<sub>ij</sub> と x の類

```

1 for (i = 1; i < m; i++) do
2   for (j = 1; j < n; j++) do
3     G(x) =  $\sum_{ij} \alpha_{ij} (\mathbf{h}(\mathbf{x}_{i1}) \cdot \mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{x}_{ij}) \cdot \mathbf{h}(\mathbf{x}))$ 
4      $\hat{j} = \arg \max_{j=1 \dots N+1} G(\mathbf{x}_{ij})$ 
5     if ( $\hat{j} \neq 1$ ) then
6        $\alpha_{ij} = \alpha_{ij} + 1$ 
7     endif
8   end
9 end

```

図4 Ranking Voted Perceptron

表1 実験データの詳細

	訓練データ	テストデータ
短縮課題の形態素数の平均	43.0	42.8
リファレンスの形態素数の平均	25.4	25.3
要約率平均 (削除率平均)	0.59(0.41)	0.59(0.41)
リファレンス個数	5	5

似度と考えてよい。よって本稿では類似度尺度を、単語ユニグラム、スキップバイグラム、トライグラム、品詞バイグラムの重み付き線形和とし、次式で表す。

$$\begin{aligned}
\mathbf{h}(\mathbf{x}_{ij}) \cdot \mathbf{h}(\mathbf{x}) = & \lambda_1 \text{unigram}(\mathbf{x}_{ij}, \mathbf{x}) \\
& + \lambda_2 \text{skipbigram}_d(\mathbf{x}_{ij}, \mathbf{x}) \\
& + \lambda_3 \text{trigram}(\mathbf{x}_{ij}, \mathbf{x}) \\
& + \lambda_4 \text{posbigram}(\mathbf{x}_{ij}, \mathbf{x}) \quad (10)
\end{aligned}$$

但し  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  は各項に与えるパラメータであり、 $\text{unigram}(\mathbf{x}_{ij}, \mathbf{x})$  は  $\mathbf{x}_{ij}$  と  $\mathbf{x}$  の単語ユニグラム一致数を表す。また  $\text{skipbigram}_d(\mathbf{x}_{ij}, \mathbf{x})$  は  $d$  個以下のスキップを許した  $\mathbf{x}_{ij}$  と  $\mathbf{x}$  の単語の組合せの一致数を表し、以下で述べる評価実験では、 $d = 4$  に設定し、4 個以下のスキップを許したスキップバイグラムを用いた。なお  $\text{trigram}(\mathbf{x}_{ij}, \mathbf{x})$  は  $\mathbf{x}_{ij}$  と  $\mathbf{x}$  の単語トライグラム一致数を表し、 $\text{posbigram}(\mathbf{x}_{ij}, \mathbf{x})$  は  $\mathbf{x}_{ij}$  と  $\mathbf{x}$  の品詞バイグラムの一致数を表す。

## 4 評価実験

### 4.1 実験データ

実験データには、ステートの重み、リンクの重みに対するパラメータを調整するための訓練データとして 200 題の短縮課題を用意した。またこれとは別の短縮課題をテストデータとしての 200 題用意した。訓練データとテストデータには人手で作成したリファレンス<sup>\*4</sup>が 1 つの短縮課題に対し、複数用意してある。なお各短縮課題にはそれぞれ指定した要約率<sup>\*5</sup>があり、リファレンスもその要約率に従って作成した。表 1 に実験データの詳細を示す。

単語重要度や言語モデルは毎日新聞 3 年分 (2000 ~ 2002 年) を用いて構築した。本稿ではまず訓練データを使い、前向き DP で用いるステート及び、リンクの最適な重みパラメータを設定する。その後、得られた最適パラメータを使い、N-best 解の出力及び優先度学習を行い、オープンテストを行う。なお、テストでは各短縮課題に対して 10 個 (N-best における N=10) の出力を行い、10 分割交差検定により評価した。

\*4 リファレンスの作成は 1 人によって行った。

\*5 本稿では入力文中の形態素数に対する出力文中の形態素数の割合とする。

### 4.2 評価指標

本稿では機械翻訳の自動評価で用いられる BLEU スコア [6] とそれを要約の自動評価用に改良された ROUGE スコア [7] を使って評価を行う。

BLEU スコアはある短縮課題でシステムが出力した短縮文  $c$  とその短縮課題に対するリファレンス文の集合  $\mathbf{R}$  との類似度を求めることによって評価するものである。以下に BLEU スコアについて示す。

$$\text{BLEU}(c, \mathbf{R}) = BP \cdot \exp\left(\sum_{n=1}^4 \frac{1}{n} \log p_n\right) \quad (11)$$

$$p_n = \frac{\sum_{n\text{gram}} \text{Count}_{\text{clip}}(n\text{gram})(c, \mathbf{R})}{\sum_{n\text{gram}'} \text{Count}(n\text{gram}')} \quad (12)$$

複数リファレンスの場合  $\text{Count}_{\text{clip}}(n\text{gram})(c, \mathbf{R})$  は次式で表される。

$$\begin{aligned}
& \text{Count}_{\text{clip}}(n\text{gram})(c, \mathbf{R}) = \\
& \min(\text{Count}(n\text{gram}), \text{Max\_Ref\_Count}(n\text{gram})) \quad (13)
\end{aligned}$$

式 (12) の  $\text{Count}(n\text{gram}')$  はシステム出力の各 ngram の数である。また式 (13) での  $\text{Count}(n\text{gram})$  はシステム出力に含まれるある語の数であり、 $\text{Max\_Ref\_Count}(n\text{gram})$  は複数リファレンスに含まれるある語の最大数である。本来、BLEU スコアではシステム出力がリファレンスよりも短かった場合、ペナルティ ( $BP$ ) を与えるが、今回は各短縮課題でリファレンスとシステム出力の長さが一致するためこのペナルティについては課さない。

ROUGE スコアはある短縮課題でシステムが出力した短縮文  $c$  とその短縮課題に対する複数リファレンス 1 つ 1 つ  $r$  との類似度を求めることによって評価するものである。以下に ROUGE スコアについて示す。

$$\text{ROUGE-N}(c, r) = \frac{\sum_{n\text{gram}} \text{Count}_{\text{match}}(n\text{gram})}{\sum_{n\text{gram}'} \text{Count}(n\text{gram}')} \quad (14)$$

式 (14) の  $\text{Count}(n\text{gram}')$  はリファレンスの各 ngram の数であり、 $\text{Count}_{\text{match}}(n\text{gram})$  はリファレンスとシステム出力での一致したある語の数である。

### 4.3 評価結果

評価実験では各短縮課題で指定された要約率に従って文短縮を行った。評価では優先度学習によってより良い短縮文が上位に再ランキングできたかを調べるため、優先度学習前後での BLEU スコア、ROUGE スコアの比較を行う。ROUGE スコアでは ROUGE-1, ROUGE-2, ROUGE-3 で評価を行った。なお複数リファレンスであるため、各短縮課題での最大 ROUGE スコアを示す。表 2, 3 に優先度学習前後での評価の比較を示す。表 2 では優先度学習前後における 1 位と 3 位以内の BLEU スコアの平均を示し、表 3 では優先度学習前後における 1 位の ROUGE スコアの平均を示す。

表 2 より 1 位の BLEU スコア平均では約 3 ポイント改善でき、3 位以内での BLEU スコア平均も改善できた。また表 3 より ROUGE による評価でも優先度学習前に比べ改善できたことが分かる。なお出力した 10-best 中最も高い BLEU スコアの順位平均を調べたところ、優先度学習前

表2 優先度学習前後での BLEU スコアによる評価の比較

	優先度学習前	優先度学習後	前後での差
1位	.440	.474	+0.034
3位以内	.437	.461	+0.024

表3 優先度学習前後での ROUGE スコアによる評価の比較

	優先度学習前	優先度学習後	前後での差
ROUGE-1	.676	.698	+0.022
ROUGE-2	.502	.526	+0.024
ROUGE-3	.375	.404	+0.029

短縮課題：

規制緩和をめぐり、与党の行政改革プロジェクトチームが20日午後、通産省、運輸省などの官房長らを相手に異例の公開討論会を行う

優先度学習前

1. 規制緩和をめぐり、プロジェクトチームが20日、相手に異例の公開討論会を行う
2. 規制緩和をめぐり、プロジェクトチームが20日、を相手に異例の討論会を行う
3. 規制緩和をめぐり、改革プロジェクトチームが20日、相手に異例の討論会を行う

優先度学習後 (学習前の順位)

1. 規制緩和をめぐり、行政改革プロジェクトチームが20日、異例の公開討論会を行う (9)
2. 規制緩和をめぐり、改革プロジェクトチームが20日、相手に異例の討論会を行う (3)
3. 規制緩和をめぐり、プロジェクトチームが20日、を相手に異例の討論会を行う (2)

図5 優先度学習前後における短縮文のランキング例

が5.4位だったのに対し、優先度学習後では3.7位となり、2位近く改善することができた。これらの結果から、優先度学習を適用したことで、より良い短縮文を上位に再ランキング出来たことが分かる。さらに表3より ROUGE-3 の評価でも優先度学習の効果が表れていることが分かる。これは優先度学習の類似度尺度に単語トライグラムまでを用いたことにより単語接続のより良いものを上位に再ランキング出来たためであると考えられる。

#### 4.4 考察

##### 短縮文のランキング例

参考として、文短縮で DP スコアによりランキングした上位3位の結果と優先度学習により再ランキングした上位3位の結果の例を図5に示す。優先度学習後の括弧内の数字は優先度学習前の順位である。優先度学習前では9位だった短縮文が優先度学習後では1位になっており大きく順位が入れ替わっていることが分かる。この例では優先度学習の効果も良く分かる。

##### リンクの重みについて

堀らは単語接続の良さを表すリンクの重みに単語トライグラム確率を用いた。それに対し、本稿では計算効率を考え単語バイグラム確率を用い、また優先度学習での類似度尺度に単語トライグラムやスキップバイグラムを用いることで単語接続がより良いものを上位に再ランキングしている。そこでリンクの重みに単語トライグラム確率を用いた場合と単語バイグラム確率を用いた場合で評価の比較を行う。なお単語トライグラム確率を用いた場合も訓練データでステートとリンクの重みの最適パラメータを設定し、得

表4 リンクの重みの違いによる評価

リンクの重み	1位の BLEU 平均
単語トライグラム	.472
単語バイグラム (優先度学習前)	.440
単語バイグラム (優先度学習後)	.474

られた最適パラメータを用いてオープンテストを行った。表4にリンクの重みの違いによる評価を示す。

この結果よりリンクの重みに単語トライグラム確率を用いた場合と単語バイグラム確率を用いた場合を比較すると優先度学習前では単語トライグラム確率を用いた方が良い評価が得られた。しかし優先度学習後ではより良い短縮文が上位に再ランキングされ、堀らの単語トライグラム確率を用いた場合に匹敵する評価が得られた。よって提案手法では計算効率を抑え、さらにオンライン型の優先度学習を用いたことで単語トライグラム確率を用いた結果に匹敵する結果を得ることが出来る。また本稿では優先度学習の類似度尺度とした単語ユニグラム、スキップバイグラム、トライグラム、品詞バイグラムを用いたが、他により適する類似度尺度を選択できればさらに良い優先度学習が行えると期待できる。

#### 5 おわりに

本稿では、既存研究をベースにオンライン型の優先度学習を適用することで、より良い短縮文を再ランキングする手法を提案した。優先度学習後では BLEU スコアによる評価が約3ポイント改善され、提案手法の有用性を確認した。さらに優先度学習後ではリンクの重みに単語トライグラム確率を用いた場合に匹敵する結果が得られ、計算効率を抑えた文短縮器を構成することが出来た。また今後の課題として優先度学習で用いる類似度尺度の選択やより細かなパラメータの設定などが挙げられる。

#### 参考文献

- [1] K. Kevin and M. Daniel. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. In *Artificial Intelligence*, vol.139, pp.91-107(2002).
- [2] 堀智織, 古井貞熙. 単語抽出による音声要約生成法とその評価. 電子情報通信学会論文誌, vol.J85-D-II, no.2, pp.200-209, (2002).
- [3] 工藤拓, 松本裕治. 相対的な係りやすさを考慮した日本語係り受け解析モデル. 情報処理学会論文誌, vol.46, no.4, pp.1082-1092, (2005).
- [4] 永田昌明. 統計的言語モデルと N-best 探索を用いた日本語形態素解析法. 情報処理学会論文誌, vol.40, no.9, pp.3420-3431, (1999).
- [5] M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and voted perceptron. *Proc. of the 40th ACL*, pp.263-270, (2002).
- [6] K. Papineni, S. Roukos, T. Ward and W. J. Zhu: BLEU: a Method for Automatic Evaluation of Machine Translation. *Proc. of the 40th ACL*, pp.311-318, (2002).
- [7] C. Lin: Looking for a Good Metrics: ROUGE and its Evaluation, *Proc. of the 4th NTCIR Workshops*, pp.1-8, (2004).