情報処理学会論文誌　データベース　Vol. 2　No. 1　1–9 (Mar. 2009)

構文情報に依存しない文短縮手法

平尾　努[†1]　鈴木　潤[†1]　磯崎秀樹[†1]

従来の文短縮手法の多くは，入力された文を構文木として表現し，その部分木を削除することで，短縮文を生成する．このようなアプローチは文法的な短縮文を生成するという観点からは理にかなっている．しかし，多くの場合，人間は構文木の刈り込みだけで短縮文を生成するわけではない．これは，構文情報に過度に依存することが，高品質な文短縮を行うための妨げとなることを示している．そこで，本稿では，構文情報を用いない文短縮手法を提案する．短縮文の言語としてのもっともらしさを構文情報を用いずに評価するため，原文と大規模コーパスから得た統計情報を組み合わせた新たな言語モデルを提案する．提案手法を文献 18) のテストセットを用いて評価したところ，自動評価指標においては，提案手法が従来法より優れていることを確認した．さらに，提案手法が日本語だけでなく英語でも有効であることも示す．

A Syntax Free Approach for Sentence Compression

Tsutomu Hirao,[†1] Jun Suzuki[†1] and Hideki Isozaki[†1]

Conventional sentence compression methods build a parse tree and then trim the tree. This approach is reasonable beacuse the compressed sentence keeps fluency. However, in many cases, reference compressions that were made by humans do not always retain syntactic structures of original sentences but they are acceptable. This implies that syntax is an impediment to achieving human-quality compression. Therefore, this paper propses a syntax free sentence compressor. As an alternative to syntactic information, we propose a novel language model that combines statistics from an original sentence and a general corpus. We conducted experimental evaluation on the test set used in Hirao, et al. [18]. The results showed that our method outperformed the conventional method in automatic metrics. Moreover, we show the effectiveness of our method for English compression.

†1 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
NTT Communication Science Laboratories, NTT Corporation

1. はじめに

自動要約技術は，人間の情報アクセスを手助けするために必須であり，近年では，特に，文字数制限が厳しい状況下で要約を生成することが求められている．高圧縮でかつ質の高い要約を生成するためには，重要文抽出技術だけではなく，抽出した文の大意を保持したまま，人間が許容可能な程度の読みやさを備えるように 1 文を要約する[⋆1]文短縮技術が必要となる．すなわち，現在の要約システムにおいて，文短縮技術は重要な役割を担っている．

短縮した文の言語としてのもっともらしさを確保するため，従来の手法は，構文情報を利用している．英語の場合，文を句構造木として表現し，その部分木を削除する（木の枝を刈る）ことで短縮文を得る手法[5),15),16)]，日本語の場合，文を句構造木の代わりに依存構造木として表現し，その部分木を削除する手法[11),12),14)] がそれぞれ提案されている．つまり，構文木を刈り込むことで文法的に正しい短縮文を得ようとしている．

これに対し，文を構文木ではなく単なる単語列としてとらえる手法も提案されている[1),2),9),10),18)]．これらの手法は，構文木の刈り込みでは削除できない単語，単語列を削除することができるが，短縮文が非文にならぬよう構文構造を素性として埋め込んでいる．

一方，人間による文短縮を観察すると，構文木の刈り込みだけで短縮を行ってはいない．2 章で詳しく述べるが，文献 18) のテストセットでは，構文木の刈り込みのみで実現できない短縮文，すなわち，依存構造木の中間ノードを削除して生成された短縮文は全体の約 87%を占めた．

こうしたことを考慮すると，構文情報に強く依存する文短縮手法は，人間の短縮（リファレンス）を再現するという立場からは，望ましくない．そこで，本稿では，構文情報を用いない文短縮手法を新たに提案する．構文情報に依存せずに短縮文の言語としてのもっともらしさを評価するため，Patched Language Model（PLM）を提案する．PLM は原文と大規模コーパスから得た統計情報を組合せた言語モデルである．また，提案手法には，構文解析器を用いないため，処理時間が短縮できる，構文解析器のエラーに依存しないという利点もあわせ持つ．

以下，2 章で，人間が文を短縮する際，原文の依存構造木の中間ノードにあたる文節，単語を積極的に削除することを示す．3 章では，PLM を提案し，構文に依存しない文短縮法を提案する．4 章で評価実験の設定ついて述べ，結果を考察する．5 章では，提案手法の英

⋆1 ただし，単語順序の変更や新たな語句の挿入は考えない．

語での有効性を議論する．6 章で，関連研究について述べる．

2. 人間による文短縮

まず，実例をあげて人間の文短縮を観察する．**図 1** に以下の原文と人間による短縮文の構文構造（依存構造）を示す．

原文　[センタ試験で]$_{\text{文節 1}}$ [公表していない]$_{\text{文節 2}}$[枝問部分の]$_{\text{文節 3}}$[配点について]$_{\text{文節 4}}$[福武が]$_{\text{文節 5}}$[推定した．]$_{\text{文節 6}}$

短縮文　[センタ試験枝問の]$_{\text{文節 1-1}}$[配点について]$_{\text{文節 4}}$[福武が]$_{\text{文節 5}}$[推定した．]$_{\text{文節 6}}$

図 1 の構文構造，すなわち依存構造は，CaboCha[6] を用いて文節間の依存構造を解析した後に Kudo らのルール[7] に従い，単語間の依存構造に変換したものである[⋆1]．図中のボックスは文節を表す．

原文における文節 4〜6 に関しては，短縮文でもそのまま依存構造が保存されている．しかし，短縮文の文節 1-1,「センタ試験枝問の」という文節は，文節 1 の一部の単語「センタ」,「試験」と文節 3 の一部「枝」,「問」,「の」を結合することで生成された新たな文節であり，原文には存在しない．このとき，依存構造木の中間ノードである文節 2 が完全に削除され，文節 1，3 の一部の単語が削除されている．原文と比較すると読みやすさ（文としての滑らかさ）はやや劣っているものの，短縮文としての役割は十分に果たす文であるし，文としても十分に文法的である．

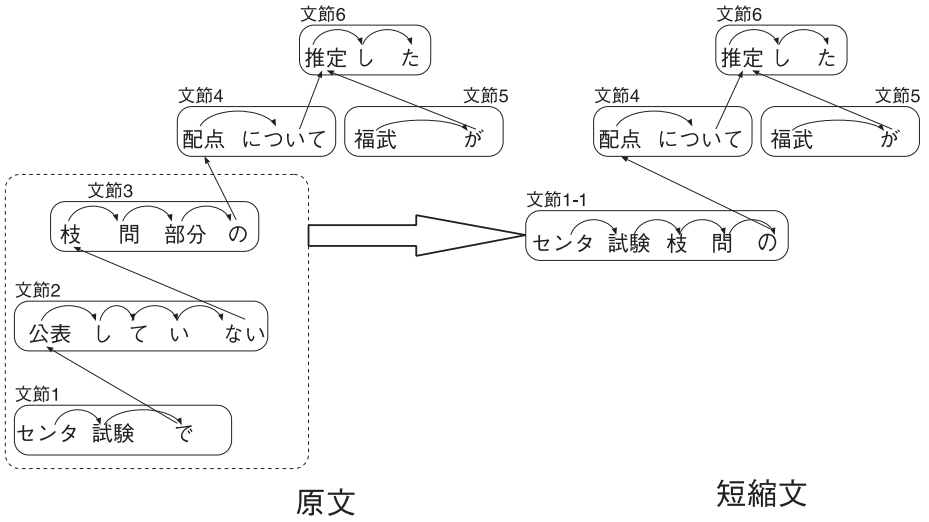

図 1　原文とその短縮文に対する依存構造の例

Fig. 1　An example of a dependency relation for a original sentence and its compressed sentence.

⋆1 文節内については左から右隣の単語に係り，文節内の最後の単語が係り先文節の主辞に係るというルール．

文献 18) で用いられたテストセットを分析したところ，5,000 リファレンス（1,000 文に対しそれぞれ 5 つのリファレンス）のうち，先の例に示したように原文の依存構造木の中間ノードである文節，単語を削除して生成されたリファレンスは，4,355 文であり，約 87%を占めた．また，中間ノードを削除することで生成されたリファレンスを少なくとも 1 つ保持する文の割合は，98.7%であった．

このように，人間が構文木の枝を刈り込むことだけで，文を短縮することは少ない．

3. 構文情報に依存しない文短縮手法

前章で観察に基づくと，文短縮手法は，依存構造木の中間ノードを積極的に削除できなければならない．よって，本稿では，文を構文木ではなく単なる単語の列と見なし，その取捨選択によって短縮文を得る．さらに，従来法が依存構造を素性として用いる理由は，短縮文の文法性，言語としてのもっともらしさを確保するためにあるが，本稿では，その代替として，新たに Patched Language Model（PLM）を提案する．

3.1 文短縮モデル

本稿では，文献 2)，18) の枠組みを利用する．つまり，N 個の単語列から M 個の部分単語列を選択する組合せ最適化問題としてとらえる．

いま，N 個の単語からなる文を $\boldsymbol{x} = x_1, x_2, \ldots, x_i, \ldots, x_N$ とする．x_j は文 $\boldsymbol{x}$ の j 番目の単語を表す．ここで，$\boldsymbol{x}$ の部分単語列，つまり，短縮文を $\boldsymbol{y} = \{y_1, y_2, \ldots, y_j, \ldots, y_M\}$ と表す．ただし，$y_j \in \{x_1, \ldots, x_N\}$ である．また，$x_0 = y_0 =$ <s>，$x_{N+1} = y_{M+1} =$ </s> とする．さらに，y_j を原文における単語のインデックスへとマッピングする関数として $I(\cdot)$ を導入する．つまり，$1 \leq I(y_j) \leq N$ である．ただし，語順の入れ替えを考えないことから，$I(y_j) < I(y_{j+1})$ となることに注意されたい．短縮文のスコアを，文献 2)，18) の手法を一般化し，以下の式 (1) で定義する．

$$f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\Lambda}) = \sum_{j=1}^{M+1} \{g(\boldsymbol{x}, I(y_j); \boldsymbol{\lambda}_g) + h(\boldsymbol{x}, I(y_j), I(y_{j-1}); \boldsymbol{\lambda}_h)\} \quad (1)$$

ただし，$\mathbf{\Lambda} = \{\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_h\}$ はパラメータベクトルである．式 (1) の右辺第 1 項は，短縮文に含まれる情報の重要度を評価し，第 2 項は短縮文の言語としてのもっともらしさを隣接する 2 つの単語によって評価している．なお，式 (1) を最大とする短縮文 $\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}; \mathbf{\Lambda})$ は，動的計画法を用いて，$O(NM)$ で得ることができる．

3.2 Patched Language Model（PLM）

$h(\boldsymbol{x}, I(y_j), I(y_{j-1}); \boldsymbol{\lambda}_h)$ には，N グラム言語モデルが採用されることが多い．一般的に，N グラム言語モデルは長短様々な長さの文を含む大量のコーパスを用いて計算される．しかし，文短縮タスクに必要とされるのは，このような一般的なコーパスから得た N グラムの確率ではなく，短縮された文に特化した N グラムの確率である．1 つの方法としては，単純に短縮文のみを大量に集め，N グラム言語モデルを構築すればよい．しかし，短縮文のみを大量に集めることは困難である．たとえば，新聞記事のヘッドラインを短縮文と見なし，集めたとしてもその量は通常の N グラム言語モデルを計算するために用意できるコーパスの量よりもはるかに小さい．よって，データスパースネスの問題が生じる．

そこで，本稿では，原文から得た統計情報と大量のコーパスから得た統計情報を組み合わせた新たな言語モデル Patched Language Model（PLM）を提案する．以下の式で定義する．

$$\mathrm{PLM}(\boldsymbol{x}, I(y_j), I(y_{j-1})) = \begin{cases} 1 & \text{if } \ I(y_j) = I(y_{j-1}) + 1 \\ \lambda_{\mathrm{PLM}}\ \mathrm{Bigram}(\boldsymbol{x}, I(y_j), I(y_{j-1})) & \\ & \text{otherwise} \end{cases} \tag{2}$$

ただし，$0 \leq \lambda_{\mathrm{PLM}} \leq 1$，Bigram(·) は，単語バイグラムの生起確率を表す．PLM では，短縮文中に出現する単語バイグラムが，原文においてもバイグラムとして出現している場合には 1，それ以外の場合には，大量のコーパスから得たバイグラム確率に基づくスコアを与える．

なお，文献 3) では，原文書とそれに対して人間が生成した要約を比較したところ，要約中の多くのバイグラムが原文書でもバイグラムとして出現していたことを報告している．文書要約と文短縮ではややタスクが異なるが，PLM はこの結果とよく合致している．

さらに，品詞バイグラムは明らかな非文を避けるためには有効である[9]．よって，品詞バイグラムを用いて最終的に $h(\boldsymbol{x}, I(y_j), I(y_{j-1}); \boldsymbol{\lambda}_h)$ を以下の式で定義する．

$$h(\boldsymbol{x}, I(y_j), I(y_{j-1}); \boldsymbol{\lambda}_h) = \mathrm{PLM}(\boldsymbol{x}, I(y_j), I(y_{j-1})) \tag{3}$$
$$+\lambda_{\mathrm{POS}(\boldsymbol{x}, I(y_j), I(y_{j-1}))} \mathrm{POS}(k(\boldsymbol{x}, I(y_j)) | k(\boldsymbol{x}, I(y_{j-1})))$$

なお，$k()$ は，単語の品詞を返す関数，POS(·) は，品詞バイグラム確率を返す関数である．品詞バイグラムに関しては，その組合せごとに異なる重みパラメータが与えられることに注意されたい．

3.3 パラメータの最適化

後述する評価実験で，文献 18) の手法と比較するため，パラメータ $\mathbf{\Lambda}$ の最適化には，MCE 学習[4] の枠組みを用いた．

いま，訓練データ集合として T が与えられ，その要素 t に対する正解短縮文 $\boldsymbol{y}_t^*$ が与えられたとすると，そのスコアは $f(\boldsymbol{x}_t, \boldsymbol{y}_t^*; \mathbf{\Lambda})$ となる．これを用い，損失関数を以下の式で定義する．

$$d(\boldsymbol{x}_t, \boldsymbol{y}_t; \mathbf{\Lambda}) = f(\boldsymbol{x}_t, \boldsymbol{y}_t^*; \mathbf{\Lambda}) - \max_{\boldsymbol{y}_t \neq \boldsymbol{y}_t^*} f(\boldsymbol{x}_t, \hat{\boldsymbol{y}}_t; \mathbf{\Lambda}) \tag{4}$$

ただし，$\boldsymbol{y}_t^* \neq \hat{\boldsymbol{y}}_t$ である．パラメータはこの損失関数を最小化するように決定すればよい．しかし，式 (4) をそのままの形で最小化することは難しいので，平滑化関数としてシグモイド関数を利用し，最小化を行う．最終的には，以下の式を最小化する問題に帰着する．

$$L(d(\boldsymbol{x}, \boldsymbol{y}; \mathbf{\Lambda})) = \sum_t^{|T|} \frac{1}{1 + \exp(-c \times d(\boldsymbol{x}_t, \boldsymbol{y}_t; \mathbf{\Lambda}))} \tag{5}$$

実際には，勾配法を用いることで最適化を行うことができる．

4. 評 価 実 験

4.1 コーパスと評価指標

提案手法の有効性を確認するため，文献 18) のテストセットを用いて評価実験を行った．テストセットには，毎日新聞の 1994 年から 2002 年までの記事から得たリード文[⋆1]が 1,000 文含まれており，各文に対して 5 つのリファレンスがある．原文の平均単語数は 42，短縮文の平均単語数は 24 である．短縮率（$= M/N$）は約 0.6 に設定されている．

評価指標としては，翻訳の自動評価指標として広くも用いられている BLEU[13]，要約の自動評価指標として広く用いられている ROUGE-N（N = 1，2，3）[8] を用いた．BLEU を用いた理由は，それがマルチリファレンスを前提として提案された評価指標だからである．

⋆1 ヘッドラインを除いた 1 番目の文．

ROUGE-N に関しては，各リファレンスごとにスコアを計算し，最大値，平均値，最小値を記録した．なお，ROUGE に関しては，文短縮を評価する指標として適切であることを示した文献は存在しない．しかし，文献 8) において，ヘッドライン生成（正確には，文書から 10 単語の要約を生成するタスク）の評価指標として，人間の評価結果との間の相関が十分高いことが報告されているので，評価指標として採用した．

また，文献 18) と同様，5 つのリファレンスのうち BLEU スコアを最大とするリファレンス（$= \mathrm{argmax}_{r \in R} \mathrm{BLEU}(r, R \backslash r)$）をパラメータ最適化のための訓練データとして用いた．R はリファレンスの集合，r はその要素である．なお，評価は 5 分割の交差検定で行った．

4.2　比較した文短縮手法

PLM の有効性を確認するため，$h(\boldsymbol{x}, I(y_j), I(y_{j-1}); \boldsymbol{\lambda}_h)$ として，通常のバイグラム確率を用いる場合，それに加え，係り受け確率を用いる場合との比較評価を以下の組合せで行った．

(a): $\mathrm{PLM}(\boldsymbol{x}, I(y_j), I(y_{j-1})) + \lambda_{\mathrm{POS}(\boldsymbol{x}, I(y_j), I(y_{j-1}))} \mathrm{POS}(k(\boldsymbol{x}, I(y_j)) | k(\boldsymbol{x}, I(y_{j-1})))$

(b): $\mathrm{PLM}(\boldsymbol{x}, I(y_j), I(y_{j-1})) + \lambda_{\mathrm{POS}(\boldsymbol{x}, I(y_j), I(y_{j-1}))} \mathrm{POS}(k(\boldsymbol{x}, I(y_j)) | k(\boldsymbol{x}, I(y_{j-1})))$
$+ \lambda_{\mathrm{DEP}} \mathrm{DEP}(\boldsymbol{x}, I(y_j), I(y_{j-1}))$

(c): $\lambda_{\mathrm{B}} \mathrm{Bigram}(\boldsymbol{x}, I(y_j), I(y_{j-1})) + \lambda_{\mathrm{POS}(\boldsymbol{x}, I(y_j), I(y_{j-1}))} \mathrm{POS}(k(\boldsymbol{x}, I(y_j)) | k(\boldsymbol{x}, I(y_{j-1})))$

(d): $\lambda_{\mathrm{B}} \mathrm{Bigram}(\boldsymbol{x}, I(y_j), I(y_{j-1})) + \lambda_{\mathrm{POS}(\boldsymbol{x}, I(y_j), I(y_{j-1}))} \mathrm{POS}(k(\boldsymbol{x}, I(y_j)) | k(\boldsymbol{x}, I(y_{j-1})))$
$+ \lambda_{\mathrm{DEP}} \mathrm{DEP}(\boldsymbol{x}, I(y_j), I(y_{j-1}))$

(a) が提案手法，(d) が文献 18) の手法である．すべての手法に対し，品詞バイグラム確率（POS(·)），単語バイグラム確率（Bigram(·)）の計算には，毎日新聞 1994 年から 2002 年までの記事を用いた．ただし，テストセットの文を含む記事は取り除いた，また，係り受け確率 DEP(·) は，工藤らの手法[17] を用いて計算した．

なお，$g(\boldsymbol{x}, I(y_j))$ については，文献 18) と同じく，文内での出現位置に基づくスコアを採用した．下記，式 (6) で定義される．

$$g(\boldsymbol{x}, I(y_j); \boldsymbol{\lambda}_g) = w(\boldsymbol{x}, I(y_j)) \left(\sum_{j=1}^{J} m_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{1}{2} \left(\frac{\mathrm{psn}(\boldsymbol{x}, I(y_j)) - \mu_j}{\sigma_j} \right)^2 \right) \right) \tag{6}$$

$\boldsymbol{\lambda}_g = \{\mu_j, \sigma_j, m_j\}_{j=1,\ldots,J}$ であり，文献 18) と同じく，$J = 2$ とした．psn(·) は，正規化した単語の出現位置を表す関数であり，以下の式で定義される．

$$\mathrm{psn}(\boldsymbol{x}, I(y_j)) = \frac{\mathrm{start}(\boldsymbol{x}, I(y_j))}{\mathrm{length}(\boldsymbol{x})} \tag{7}$$

表 1　日本語テストセットに対する評価結果
Table 1　Results obtained from Japanese corpus.

Label		(a)	(b)	(c)	(d)	TT
BLEU		**.679***	.632	.617	.669	.598
ROUGE-1	max	**.790**	.781	.776	**.790**	.769
	avg	.690	.687	.686	**.694**	.667
	min	.589	.591	.591	**.594***	.563
ROUGE-2	max	**.670**	.638	.625	.665	.635
	avg	**.540**	.510	.500	.534	.498
	min	**.413**	.383	.374	.406	.363
ROUGE-3	max	**.587***	.546	.529	.577	.544
	avg	**.435***	.397	.386	.425	.390
	min	**.292***	.256	.247	.280	.244

(a)：提案手法　(d)：[平尾 07] [18]　TT：依存構造木の刈り込み
*は，(a) と (d) との間に有意水準 5%で差があることを示す．
検定手法は Wilcoxon の符号付順位和検定．

length($\boldsymbol{x}$) は，原文の総文字数，start($\boldsymbol{x}, I(y_j)$) は，原文の先頭からの y_j までの累積文字数を表す．なお，$w(\boldsymbol{x}, I(y_j))$ は，y_j の品詞が自立語の場合には，IDF 値，それ以外の場合にはゼロではない小さい値を与える．

上記に加え，依存構造木の刈り込みよる手法（以下，TT）との比較も行った．CaboCha を用い，依存構造木を構築した後，ルートから順に指定の単語数を満たすまで，文節を選択した．ただし，文節の途中で指定の単語数になった場合には，文節の後ろから優先して単語を選択した．この手法は，依存構造木の刈り込みのベースラインであり，刈り込みによる上限を示したものでないことに注意されたい．刈り込む部分木の優先順位を決定することはそれ自体が研究テーマであるため，本稿では，単純なルールによって刈り込むとどの程度の精度を得ることができるかを示すために用いた．

4.3　実験結果と考察

表 1 に各手法の評価結果を示す．表より提案手法が，ROUGE-1 スコア以外で，最も良いスコアであり，BLEU，ROUGE-3 に関しては，文献 18) の手法に対し，統計的に有意な差で優れている．この結果は，提案手法がリファレンスに出現するトライグラム，4 グラムのような長い N グラムを他の手法と比較して，より多く保持できていることを示している．すなわち，よりリファレンスに近い短縮が実現できていることを示している．また，TT の成績が悪いことから，依存構造木を単純なルールで刈り込むだけでは，リファレンスと乖離した短縮文になることを示している．

表 2　人間による主観評価
Table 2　The results of human evaluation.

	文法	内容
(a)	4.06	3.34
(d)	4.07	3.26

次に，構文情報の有効性について議論する．構文情報を用いない (c) は，ROUGE-1 では，それを用いる (b) とほぼ同等であり，TT には勝っている．しかし，ROUGE-2，3 と評価する N グラムを長くしていくと，(b) との差は広がり，TT との差は縮まる傾向にある．ROUGE-3 においては，TT よりも劣る．つまり，局所的にはリファレンスと似ているが，文全体ではそれに及ばない短縮文となっている可能性が高い．一方，構文情報を追加した (d) は，すべての指標で (c) よりも優れており，TT と比較しても十分良い．これらの結果から，大規模コーパスから得た単語バイグラムによる言語モデルのみでは，短縮文の文としてのもっともらしさを評価することが難しく，より良い短縮を実現するには，構文情報が必要であることが分かった．

しかし，PLM に構文情報を導入すると BLEU，ROUGE ともに成績は下がった．日本語の場合，Kudo らのルール[7] に従って，文節間の係り受けを単語間の係り受けに変換すると，多くの単語は高い確率で右隣の単語に係ることとなる．一方，PLM では，原文でのバイグラムが短縮文に出現したときに高いスコアを与える．つまり，原文でのバイグラムに高いスコアを与えるという意味で，両者はよく似た性質を持つ素性といえる．よって，相関の高い素性を追加したことにより，素性数は増えたものの事例が持つ情報量は増加しないため，パラメータの最適化時に過学習を起こしたと考える．

さらに，人間による主観評価を (a) と (d) を対象として行った．6 名の被験者が，テストセットからランダムに選択した 100 文に対し，短縮文の文法性，内容としての適切さをそれぞれを 5 段階（5 が最も良い）で評価した．その平均値を**表 2** に示す．また，5 段階の各スコアの頻度分布を**図 2** に示す．文法性に関してはほぼ同等，内容としての適切性ではやや (a) が良い．また，評価「5」を得た数は双方の指標で (a) が多い．これらの結果より，提案手法は，構文解析を用いていないにもかかわらず (d) と少なくとも同等以上の精度であると考える．

また，**図 3** に (a)，(d)，TT による短縮文の例を示す．1 文目の例では，「自民党高知県連」が「開き」に係り，「開き」が「決めた」に係るという依存構造であるが，「決めた」の動作主体は「自民党高知県連」である．人間はこうした文の意味構造をよくとらえており，

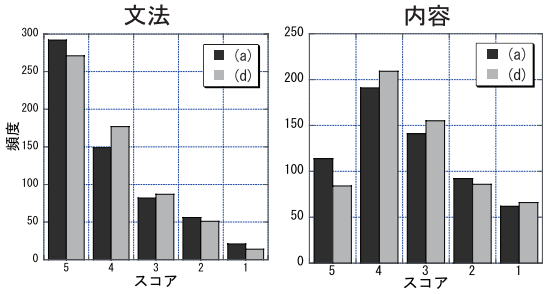

図 2　スコアの頻度分布
Fig. 2　The frequency distribution for each score.

中間ノードである「開き」を削除し，「自民党高知県連」が「決めた」に係るように文法的な短縮文を生成している．一方，TT は，依存構造木の枝を刈り込むことしかできないので，「自民党高知県連」を残し，かつ，「開き」を削除することはできない．よって，文法的ではあるが，文の大意を保持するという観点からは良い短縮にならない．(d) は，TT ほど構文に対する強い制約がないため，より柔軟な短縮ができている．しかし，「自民党」が抜けているため，短縮文の大意を保持できているとはいえない．2 文目の例では，どの手法とも短縮文として文法的であるが，(a) は，文の主語が他のものと比較して劣っている．TT は，主語に問題はないが，「民活インフラ事業」の連体修飾節である「発展途上国の」が抜けて落ちており，やや焦点がぼやけている．(d) は依存構造木の刈り込みではあるが，TT よりも柔軟な刈り込みであり，リファレンスと一致している．3 文目の例では，リファレンスが依存構造木の刈り込みによって生成されており，TT と一致している．(a) は，言語としてのもっともらしさは十分に確保しているが，原文の大意を保持するという観点から，短縮文としては不適切である．(d) は「制度改革が動き出す」という原文の核となる部分は保持しているが，意味構造がやや破綻している．

さらに，提案手法は，構文解析を用いないことで，従来手法より処理時間の短縮が期待できる．そこで，(a) と (d) の処理速度を比較した．計算機は，CPU が Intel© Core™ 2 Extreme QX9650（3.00 GHz），メモリ 8 G バイトである．テストセット 1,000 文を処理するのにかかった時間は，下記のとおりである．

(a)　22.14 秒（45.2 文/秒）
(d)　59.45 秒（16.8 文/秒）

提案手法は，従来法より 2.7 倍程度高速である．よって，大量のデータを処理する場合や

原文	自民党高知県連は 19 日、常任総務会を開き、今秋予定の党総裁選の前倒しを党本部に求めることを決めた。
リファレンス	自民党高知県連は党総裁選の前倒しを本部に求めることを決めた。
(a)	自民党高知県連は総裁選の前倒しを党本部に求めることを決めた。
(d)	高知県連は党総裁選の前倒しを党本部に求めることを決めた。
TT	開き、今秋党総裁選の前倒しを党本部に求めることを決めた。
原文	日本輸出入銀行、海外経済協力基金、商社、都銀などで構成する民活インフラ研究会は 10 日、開発途上国の民活インフラ事業を活発化させる提言をまとめ、大蔵省に提出した。
リファレンス	民活インフラ研究会は開発途上国の民活インフラ事業を活発化させる提言をまとめ、大蔵省に提出した。
(a)	日本輸出入銀行、海外経済協力基金は民活インフラ事業を活発化させる提言をまとめ、大蔵省に提出した。
(d)	民活インフラ研究会は開発途上国の民活インフラ事業を活発化させる提言をまとめ、大蔵省に提出した。
TT	民活インフラ研究会は 10 日、民活インフラ事業を活発化させる提言をまとめ、大蔵省に提出した。
原文	住宅金融専門会社や金融機関の破たんにともない、存在意義が問われている会計士監査の制度改革が動き出すことになった。
リファレンス	存在意義が問われている会計士監査制度改革が動き出すことになった。
(a)	住宅金融専門会社や金融機関の破たんにともない、存在意義が問われた。
(d)	住宅金融専門会社や存在意義が問われ制度改革が動き出すことになった。
TT	存在意義が問われている会計士監査の制度改革が動き出すことになった。

図 3　評価データに対する短縮文の例
Fig. 3　Example compressions for the evaluation data.

リアルタイムでの処理が要求される場合に有効である．

5. 英語文短縮における提案手法の有効性

構文情報を必要としない提案手法が，文法的に厳格な英語でどのような振舞いを示すかを確認するため，文献 5) の Ziff-Davis コーパスを用いた評価実験を行った．1,035 文が訓練，32 文がテストデータとして与えられている．文献 5) で提案された構文木の刈り込みによる 2 手法，決定木に基づく短縮手法（以下，Dtree），Noisy Channel に基づく短縮手法（以下，Noisy）との比較を行った．リファレンスは 1 つしか用意されていないので，評価指標には ROUGE のみを用いた．ただし，各手法によって，文短縮率が異なるので，ROUGE の精度，再現率，F 値で評価した．提案手法は，リファレンスと同じ長さの短縮率を設定した．なお，各手法とも短縮率が異なるため厳密な比較が難しいことに注意されたい．**表 3** にその結果を示す．

表 3　英語テストセットに対する評価結果
Table 3　Results obtained from English corpus.

Label		(a)	Dtree	Noisy
ROUGE-1	Rec.	–	.792	.840
	Prec.	–	.748	.671
	Fsc.	.778	.704	.705
ROUGE-2	Rec.	–	.694	.693
	Prec.	–	.631	.562
	Fsc.	.714	.628	.593
ROUGE-3	Rec.	–	.592	.561
	Prec.	–	.533	.453
	Fsc.	.627	.533	.476
Compression rate		.533	.572	.704

表より，Dtree は Noisy よりも短縮率が低い．ROUGE-1 では，再現率で Noisy に劣るものの，ROUGE-2，ROUGE-3 では再現率，精度ともに Noisy を上回っている．これらに対し，提案手法は構文情報を用いていないにもかかわらず，どの評価指標においても成績が最も良い．ただし，提案手法と Dtree との間に統計的に有意な差は見られなかった．これは，サンプル数が少ないことが影響していると考える．以上より，提案手法は，日本語のような文法的制約が緩い言語だけでなく，英語のような文法的に厳格な言語であって有効に働くことが分かった．

各手法の短縮例を**図 4** に示す．1 文目の例では，提案手法はリファレンスを再現できている．Noisy は 1 単語しか削除できておらず，Dtree は多くの単語を削除できたものの原文の大意を保持するような短縮にはなっていない．2 文目の例では，Noisy は短縮ができていない．提案手法と Dtree は文法的な短縮文となっているが，どちらが良い短縮かはそれを読む人間の背景知識に依存するであろう．一方，3 文目の例では，どの手法でも文法的な短縮であるが，内容的には Noisy が原文の大意をよく保持しており，リファレンスとも近い．

原文	Many debugging features , including user-defined break points and variable-watching and message-watching windows , have been added .
リファレンス	Many debugging features have been added .
提案手法	Many debugging features have been added .
Dtree	Many debugging features .
Noisy	Many debugging features , including user-defined points and variable-watching and message-watching windows , have been added .
原文	Examples for editors are applicable to awk patterns , grep and egrep .
リファレンス	Examples are applicable to awk patterns , grep and egrep .
提案手法	Examples for editors are applicable to awk grep and egrep .
Dtree	Examples for editors are applicable to awk patterns .
Noisy	Examples for editors are applicable to awk patterns , grep and egrep .
原文	* 68000 sweden ab of uppsala , sweden , introduced the teleserve , an integrated answering machine and voice-message handler that links a macintosh to touch-tone phones .
リファレンス	68000 sweden ab introduced the teleserve integrated answering machine and voice-message handler .
提案手法	* 68000 sweden ab of uppsala links a macintosh to touch-tone phones .
Dtree	* 68000 sweden ab of uppsala
Noisy	* 68000 sweden ab of uppsala , sweden , introduced the teleserve , an integrated answering machine and voice-message handler

図 4　英語の評価データに対する短縮文の例
Fig. 4　Example compressions for the English evaluation data.

Dtree は，1 文目の例と同じく，原文の大意を保持はしていない．提案手法は，短縮によって原文とは異なる意味が生成されてしまっている．

6. 関 連 研 究

英語を対象とした文短縮では，文を句構造木として表現し，原文と短縮文を対応付けた後，その部分木の削除（刈り込み）ルールを学習する手法が提案されている．文献 5) では，確率モデルと決定木，文献 16) では最大エントロピーモデルによりこれを学習している．さらに，文献 15) では，文献 5) の確率モデルに対し，教師なし，半教師あり学習の枠組みも導入されている．

日本語の場合，文を句構造木として表現するのではなく，文節をノードとする依存構造木として表現し，それを刈り込む手法が提案されている．文献 14) では，SVM，文献 11) では CRF を用いた刈り込み手法が提案されている．文献 12) では，依存構造の整合性を保ちつつ，文節重要度の総和を最大とする部分木を選択する手法が提案されている．

構文木の刈り込みによる文短縮は，構文解析のエラーがない限り，文法的であることが保証されるという利点がある．しかし，日本語の場合，2 章で観察したように，人間は構文木の刈り込みのみで短縮を行ってはいないので，リファレンスを再現するために適切な手法とはいえない．

一方，文を木構造ではなく単語列と見なし，単語の選択により短縮文を生成する手法も提案されている[1),2),9),10),18)]．これらの手法では，依存構造木における中間ノードを削除する可能性を持っている．リファレンスを再現するという立場からは，これは，構文木の刈り込みによるアプローチに対する大きな利点といえる．ただし，短縮文の言語としてのもっともらしさをバイグラム，トライグラム程度の言語モデルのみで評価した場合，非文となる可能性が非常に高い（たとえば，文献 5) のベースライン手法）ため，構文構造を素性として埋め込んでいる．文献 2)，18) では，依存構造の情報，文献 9) では，依存構造に加え，句構造の情報も導入されている．さらに，文献 1) では，あらかじめ，文の主語，動詞，目的語を特定するという前処理が導入されている．

提案手法は，単語選択モデルではあるが，構文解析を用いずにそれを用いる手法と同等以上に高精度，高速である点で従来法とは大きく異なる．

7. ま と め

本稿では，構文情報を必要としない文短縮手法を提案し，構文情報の代替として，PLM を提案した．日本語，英語のテストセットを用いて評価実験を行った結果，日本語においては，構文情報を用いた従来法より，少なくとも同等以上の精度であることを確認し，従来法より高速であることも確認した．さらに，日本語よりも文法的に厳格な英語に対しても提案手法の有効性を示した．

本稿での貢献は，以下の 4 点である．

- 日本語の場合，多くの短縮文において，依存構造木の中間ノードにあたる文節，単語が

積極的に削除されていたことを示した．

- 短縮文のもっともらしさを評価するための新たな言語モデルである PLM を提案した．また，これを用いることで構文情報を用いなくても従来手法と比較して少なくとも同等以上の精度を持つ文短縮手法を実現した．
- 構文解析器を用いないことで，従来手法よりも処理時間を短縮できた．
- 提案手法が，構文的に厳格な英語においても有効に働くことを文献 5) のテストセットを用いて示した．

謝辞　英語の文短縮コーパスを提供してくださった，南カリフォルニア大学の Kevin Knght 博士，Daniel Marcu 博士に感謝いたします．

参 考 文 献

1) Clarke, J. and Lapata, M.: Models for Sentence Compression: A Comparison across Domains, Training Requirements and Evaluation Measures, *Proc. 21st COLING and 44th ACL*, pp.377–384 (2006).
2) Hori, C. and Furui, S.: A New Approach to Automatic Speech Summarization, *IEEE Trans. Multimedia*, Vol.5, No.3, pp.368–378 (2003).
3) Jing, H. and McKeown, K.: The Decomposition of Human-Written Summary Sentences, *Proc. 22nd SIGIR*, pp.129–136 (1999).
4) Juang, B.H. and Katagiri, S.: Discriminative Learning for Minimum Error Classification, *IEEE Trans. Signal Processing*, Vol.40, No.12, pp.3043–3053 (1992).
5) Knight, K. and Marcu, D.: Summarization Beyond Sentence Extraction, *Artificial Intelligence*, Vol.139, No.1, pp.91–107 (2002).
6) Kudo, T. and Matsumoto, Y.: Japanese Dependency Analysis using Cascaded Chunking, *Proc. CoNLL*, pp.63–69 (2002).
7) Kudo, T. and Matsumoto, Y.: A Boosting Algorithm for Classification of Semi-Structured Text, *Proc. EMNLP*, pp.301–308 (2004).
8) Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries, *Proc. Workshop on Text Summarization Branches Out*, pp.74–81 (2004).
9) McDonald, R.: Discriminative Sentence Compression with Soft Syntactic Evidence, *Proc. 11th EACL*, pp.297–304 (2006).
10) Nomoto, T.: Discriminative Sentence Compression with Conditional Random Fields, *Information Processing and Management*, Vol.43, No.6, pp.1571–1587 (2007).
11) Nomoto, T.: A Generic Sentence Trimmer with CRFs, *Proc. ACL-08: HLT*, pp.299–307 (2008).
12) Oguro, R., Sekiya, H., Morooka, Y., Takagi, K. and Ozeki, K.: Evaluation of a Japanese sentence compression method based on phrase significance and inter-phrase dependency, *Proc. TSD 2002*, pp.27–32 (2002).
13) Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: BLEU: A Method for Automatic Evaluation of Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistic* (*ACL*), pp.311–318 (2002).
14) Takeuchi, K. and Matsumoto, Y.: Acquisition of Sentence Reduction Rules for Improving Quality of Text Summaries, *Proc. 6th NLPRS*, pp.447–452 (2001).
15) Turner, J. and Charniak, E.: Supervised and Unsupervised Learning for Sentence Compression, *Proc. 43rd ACL*, pp.290–297 (2005).
16) Unno, Y., Ninomiya, T., Miyao, Y. and Tsujii, J.: Trimming CFG Parse Trees for Sentence Compression Using Machine Learning Approach, *Proc. 21st COLING and 44th ACL*, pp.850–857 (2006).
17) 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).
18) 平尾　努，鈴木　潤，磯崎秀樹：識別学習による組合せ最適化問題としての文短縮手法，人工知能学会論文誌，Vol.22, No.6A, pp.574–584 (2007).

(平成 20 年 9 月 18 日受付)
(平成 20 年 12 月 23 日採録)

(担当編集委員　関 洋平)

平尾　努（正会員）

1995 年関西大学工学部電気工学科卒業．1997 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．同年（株）NTT データ入社．2000 年より NTT コミュニケーション科学基礎研究所に所属．博士（工学）．自然言語処理の研究に従事．言語処理学会，ACL 各会員．

鈴木　　潤（正会員）

1999 年慶應義塾大学理工学部数理科学科卒業．2001 年同大学院理工学研究科計算機科学専攻修士課程修了．同年日本電信電話株式会社入社．現在，NTT コミュニケーション科学基礎研究所に所属．博士（工学）．主として自然言語処理，機械学習に関する研究に従事．ACL，言語処理学会各会員．

磯崎　秀樹（正会員）

1983 年東京大学工学部計数工学科卒業．1986 年同工学系大学院修士課程修了．同年日本電信電話（株）入社．1990〜1991 年スタンフォード大学ロボティクス研究所客員研究員．現在，NTT コミュニケーション科学基礎研究所知識処理研究グループリーダ．博士（工学）．平成 15 年度情報処理学会論文賞・山下記念研究賞受賞．人工知能・自然言語処理の研究に従事．電子情報通信学会，人工知能学会，言語処理学会，ACL 各会員．