社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2009－ＦＩ－ 93 (3)
2009－ＮＬ－189 (3)
2009／1／22

重要文抽出と文圧縮を組み合わせた新たな抽出的要約手法

富田 紘平 † 高村 大也 ‡ 奥村 学 ‡
† 東京工業大学 総合理工学研究科
〒 226-8503 横浜市緑区長津田町 4259
tomita@lr.pi.titech.ac.jp
‡ 東京工業大学 精密工学研究所
〒 226-8503 横浜市緑区長津田町 4259
{takamura, oku}@pi.titech.ac.jp

重要文抽出と文圧縮を同時に行う新しい抽出的要約モデルを提案する．本手法は，重要文抽出に比べて表現能力が高いため，重要文抽出より適切な要約を出力することが期待できる．定式化には整数計画法を用いる．重要文抽出における過去の最も良い要約システムを，ROUGE-1 において大きく上回ることを確認した．

キーワード：自動要約，整数計画法

A New Approach of Extractive Summarization Combining Sentence Selection and Compression

Kohei Tomita† Hiroya Takamura‡ Manabu Okumura‡
†Tokyo Institute of Technology,
Interdisciplinary Graduate School of Science and Engineering
4259 Nagatsuta Midori-ku Yokohama,
JAPAN, 226-8503
‡Tokyo Institute of Technology,
Precision and Intelligence Laboratory
4259 Nagatsuta Midori-ku Yokohama,
JAPAN, 226-8503

We propose a novel extractive summarization model which performs sentence selection and compression at the same time. Because of its high expression ability, our model is expected to generate more appropriate summaries. Our model uses integer linear programming. Our model outperforms the previous best system on the ROUGE-1 metric.

Keywords: text summarization, integer linear programming

1 序論

本稿では，要約全体の最適性に重点を置いた，抽出的自動要約に対する新しいアプローチを提案する．自動要約とは，文書ないし文書群が与えられたとき，その内容を簡潔に表した要約を作成するタスクである．自動要約の手法には抽出的な手法と概要的 (abstractive) な手法がある．さらに，現在の抽出的自動要約は，抽出のソースとそれから抽出する言語単位 (以下抽出単位と呼ぶ) により二つに分類される．文書を対象にした重要文抽出と，文を対象にした文圧縮である．

重要文抽出は，文書を対象にした，文を抽出単位とする要約の手法である．重要文抽出の典型的な手法では，文に対してスコアを付け，文集合のスコアを集合に含まれる文のスコアの合計として，指定の要約長に収まる，最大のスコアを持つ文集合を要約として出力する．重要文抽出の手法は他にも提案されており，例えば Yih ら (2007) では，単語に対してスコアを付け，文集合のスコアを文集合に含まれる単語のスコアの合計として，指定の要約長に収まる，最大のスコアを持つ文集合を要約として出力する．どちらの手法にしても，重要文抽出に分類される手法においては，自動要約という問題を最大のスコアを持つ文集合を探す問題として解いてきた．文法性が維持されることから，重要文抽出は自動要約の研究初期から行われてきた．しかし，文を抽出単位にしているため，文のスコアが高ければ，たとえ文中に冗長な表現があったとしても要約に含まれてしまう問題点があった．一方，文圧縮は文を対象にした，単語またはフレーズを抽出単位とする要約の手法である．この手法は文法性を維持しながら要約を行わなければならない．この手法は自動要約の比較的新しい研究分野である．事実，文圧縮という語がタスクの名称として使われたのは，Knight ら (2000) の研究が初めてである．

重要文抽出と文圧縮は排他的なものではない．それどころか，文圧縮により文中の冗長な表現を取り除くことが出来るため，ここ数年の自動要約の研究においては，重要文抽出と文圧縮を逐次的に利用することが多くなってきている．例えば，Conroy(2005) は，重要文抽出の前処理として文

圧縮を行っている．また，Yih ら (2007) は圧縮前の文と圧縮後の文を入力として重要文抽出を行っている．

ただし，逐次的に利用する手法では，それぞれ重要文抽出，文圧縮という要約の要素技術を個々に扱っている．そのため，文圧縮はあくまで文にとって最適な圧縮文を提供するに過ぎず，要約全体にとって最適な文を提供しているとは限らない．そこで，要約全体の最適性に応じて，不要な文や不要なフレーズの除去を同時に行うことができる方法として，重要文抽出と文圧縮を同時に解く手法というものを考えたい．

近年，自動要約のタスクを整数計画問題として解くケースが増えている．Yih ら (2007) と高村ら (2008) は重要文抽出を最大被覆問題として定式化し，特に高村らはこれを整数計画問題の形で表現することで解いている．Clark ら (2008) は文圧縮を整数計画問題として定式化して解いている．また，概要的な手法ではあるが，Filippova ら (2008) も整数計画問題を用いている．整数計画問題は表現能力が高く，また，他の整数計画問題と結合しやすいという利点がある．そこで，本稿では，高村らが提案した重要文抽出を解く整数計画問題と，Clark らが提案した文圧縮を解く整数計画問題の双方の拡張となる整数計画問題を定式化することにより，重要文抽出と文圧縮を同時に解くことを可能にする，新しい抽出的要約手法を提案する．

2 整数計画問題

本稿では，抽出的な自動要約を整数計画問題に定式化することで解く．そこで，本節で整数計画問題について触れる．整数計画問題 (整数線形計画問題とも言う) とは最適化問題の一種であり，線形計画問題の決定変数を整数に限定した問題である．一般には以下のように定式化される:

$$\max_{\boldsymbol{x}}: \quad \boldsymbol{c}^T\boldsymbol{x} \tag{1}$$

$$subj.to: \quad \boldsymbol{A^1x} = \boldsymbol{b^1} \tag{2}$$

$$\boldsymbol{A^2x} \leq \boldsymbol{b^2}. \tag{3}$$

$\boldsymbol{x}$ を決定変数と呼ぶ．ベクトル $\boldsymbol{c}$, $\boldsymbol{b^1}$, $\boldsymbol{b^2}$, 行列 $\boldsymbol{A^1}$, $\boldsymbol{A^2}$ はパラメータである．最大化させたい式 (1) をスコア関数，決定変数が満足しなければならない式 (2) と式 (3) を制約と呼ぶ．整数計画問題が取り扱う問題は，制約を満足する中で最大のスコアを与える決定変数 $\boldsymbol{x}$ を探すことである．

3 定義

本稿では，後々の議論のために必要となる変数の定義を示す．

- m:要約の対象となる文の数
- $s_i(1 \leq i \leq m)$:要約の対象となる文
- $n_i(1 \leq i \leq m)$:文 s_i 中のトークンの数
- $w_{i,j}(1 \leq i \leq m, 1 \leq j \leq n_i)$:文 s_i 中の j 番目のトークン
- V:単語の集合
- $MARK$:記号の集合 (ピリオド，カンマ，コロン，セミコロン，アポストロフィ)
- $\xi_i \in \{0,1\}(1 \leq i \leq m)$:文 s_i が要約中に含まれるか否かを表現する決定変数
- $\delta_{i,j} \in \{0,1\}(1 \leq i \leq m, 1 \leq j \leq n_i)$:トークン $w_{i,j}$ が要約中に含まれるか否かを表現する決定変数
- $\theta_v \in \{0,1\}(v \in V)$:単語 v が要約中に含まれるか否かを表現する決定変数

決定変数 ξ_i, $\delta_{i,j}$, θ_v に関しては，それぞれ対応する文 s_i，トークン $w_{i,j}$，単語 v が要約中に含まれるとき，その値を 1 とし，そうでないとき 0 とする．

4 重要文抽出の定式化

提案手法は，高村ら (2008) の手法をベースにしている．彼らの手法は，重要文抽出を最大被覆問題として解いており，その過程で，整数計画問題として表現している．彼らのモデルは以下のように定式化される:

$$\max: \quad \lambda \sum_{v \in V} \theta_v R(v) + \sum_{i=1}^{m} \xi_i (\sum_{v \in V} R(v) a_{i,v}) \tag{4}$$

$$subj.to: \quad \forall v \sum_{i=1}^{m} a_{i,v} \xi_i \geq \theta_v \tag{5}$$

$$\sum_{i=1}^{m} \xi_i n_i \leq K. \tag{6}$$

$a_{i,v}$ は，s_i に単語 v が含まれているかを表すバイナリ変数である．K は要約長の上限である．$R(v)$ は，単語 v に対するスコアであり，ロジスティック回帰を用いて学習した位置情報を考慮したスコアである．このスコアは Yih ら (2007) が提案した．本稿ではこのスコアは採用していない．式 (4) の第一項は，要約のスコアを要約中に含まれる単語のスコアの和とすることを表している．式 (4) の第二項は，文のスコアを文中に含まれるトークンのスコアとし，要約のスコアを要約中に含まれる文のスコアとすることを表している．式 (5) は，単語と文の整合性を保つための制約である．式 (6) は，要約長に上限を与える制約である．

5 文圧縮の定式化

提案手法は，Clark ら (2008) の手法をベースにしている．そこで，Clark らの手法について簡単に

述べる. 彼らは次の式で圧縮文のスコアを求めている:

$$\sum_{j=1}^{n_i} \delta_{i,j} I(w_{i,j}). \tag{7}$$

そして, 制約を満たす中で最大のスコアをもつ圧縮文を解としている.

スコア関数 $I(w_{i,j})$ は, Hori ら (2004) が提案した節の深さを考慮した tf · idf を用いている. このスコアは

$$I(w_{i,j}) = \frac{l_{i,j}}{N_i} f_v \log \frac{F}{F_v} \tag{8}$$

と表される. $l_{i,j}$ はトークン $w_{i,j}$ が属する節の数とする. $N_i = max_j l_{i,j}$ は文 s_i 中のトークンが属する節の数の中で最大の値を表す. f_v を要約の対象となる文中の単語 v のトークンの頻度とする. F_v をコーパス中の単語 v のトークンの頻度とする. F をコーパス中の全てのトークンの頻度とする. このスコアは tf · idf に似ているが, 深いネストの位置にある節に含まれるトークンほど高いスコアとなる. なお, 内容語 (名詞, 動詞, 形容詞) 以外のトークンとストップワードのスコアは 0 とする.

制約は以下の通りである. なお, 付記している係り受け関係や品詞は, 今回構文解析に利用した RASP のものである.

修飾子制約

修飾関係にあるトークンのペアに対し, 修飾先のトークンが要約に含まれず, かつ, 修飾元のトークンが要約に含まれる場合, 係り受け関係が文の中で切れてしまう. そのため, これを禁止する制約を設ける. すなわち, トークン $w_{i,j}$ が, トークン $w_{i,k}$ を修飾 (係り受け関係:DET, NCMOD, XMOD, CMOD, TA, PMOD) するとき, 制約

$$\delta_{i,j} \geq \delta_{i,k} \tag{9}$$

を追加する. このとき, $\delta_{i,k}$ が 0 になると, $\delta_{i,j}$ も 0 になる. つまり, 修飾先のトークン $w_{i,k}$ が要約に含まれないならば, 修飾元のトークン $w_{i,j}$ も要約には含まれない.

ただし, 修飾元のトークンが'not' または所有格であるとき, 修飾元のトークンが要約に含まれないだけで, 意味が変わってしまう. これらのトークンに対しては, 異なる制約を設ける. トークン $w_{i,j}$='not' または所有格 (係り受け関係のサブタイプ:poss) であるとき, 式 (9) の代わりに制約

$$\delta_{i,j} = \delta_{i,k} \tag{10}$$

を追加する.

文型制約

動詞に対し主語または目的語の関係にあるトークンが要約に含まれないと, 構文が崩れる. そこで, これらのトークンに対して制約を持たせる. すなわち, トークン $w_{i,j}$ が, 動詞 $w_{i,k}$ の主語または目的語 (係り受け関係:NCSUBJ, XSUBJ, CSUBJ, DOBJ, OBJ2, OBJ, IOBJ) であるとき, 制約

$$\delta_{i,j} = \delta_{i,k} \tag{11}$$

を追加する.

節句制約

節や句の標識 (that, where など) が無いと節や句の役割や係り受け先が不明になる. そのため、節中のトークンが要約に含まれている限り, 標識が要約に含まれるようにする. すなわち, 節の標識 $w_{i,k}$ と節中のトークン $w_{i,j_1}, ..., w_{i,j_l}$ に対し, $1 \leq h \leq l$ の各 h において, それぞれ

$$\delta_{i,j_h} \leq \delta_{i,k} \tag{12}$$

を追加する.

一方で, 節中のすべてのトークン (記号を除く) が要約に含まれない場合, 節の標識が要約に含まれてはならない. すなわち, 節の標識 $w_{i,k}$ と節中のトークン $w_{i,j_1}, ..., w_{i,j_l}$ に対し, 制約

$$\sum_{h=1; w_{i,j_h} \notin MARK}^{l} \delta_{i,j_h} \geq \delta_{i,j_1} \tag{13}$$

を追加する.

対象となる節は, that, wh-から始まる関係詞節 (品詞:S/np-wh_vp, S/pp-wh_s, S/comp), after, because などの接続詞節 (T/frag), 副詞句 (品詞:PP/p1) である.

等位接続詞制約

トークン $w_{i,k}$ とトークン $w_{i,l}$ が等位接続詞 $w_{i,j}$ で接続されている (CONJ) とき, 制約

$$\delta_{i,j} = \delta_{i,k} \delta_{i,l}. \tag{14}$$

を加える. なお, この制約には不備がある. 例えば「A, B, and C」のように三語以上が等位接続詞で結ばれる場合, C を要約から取り除くと, 制約により and も取り除かれる. しかし, A, B が要約から取り除かれるとは限らない. よって「A, B」というフレーズが要約に含まれる可能性が残る. これでは構造を維持しているとは言えない.

要約長制約

要約長を制御できるように制約を設ける．K 語以内の要約を作成するとき，以下の制約を追加する:

$$\sum_{j=1}^{n_i} \delta_{i,j} \leq K. \tag{15}$$

6 重要文抽出と文圧縮を組み合わせた自動要約の定式化

本節では，提案手法を説明する．提案手法では，複数のスコア関数と制約を定義し，これらを組み合わせることでモデルを構成する．スコア関数を組み合わせるときは，線形補間を行い，新たにスコア関数を構成する．

本稿で提案する重要文抽出と文圧縮を同時に行うモデルは，トークンスコア，単語スコア，トークンと文の整合性制約，トークンと単語の整合性制約，構文制約の 2 つのスコアと 3 つの制約を結合したものである．なお，ベースラインの一つとして示す重要文抽出のみを行うモデルは，トークンスコア，単語スコア，語と文を同一視する制約，トークンと単語の整合性制約の 2 つのスコアと 2 つの制約を結合したものである．

6.1 トークンスコア関数

トークンに対してスコアを付与し，要約のスコアを要約に含まれるトークンのスコアの合計とするスコア関数をトークンスコア関数と呼ぶ．トークンスコア関数は以下のように表される:

$$\sum_{i=1}^{m} \sum_{j=1}^{n_i} \delta_{i,j} I(w_{i,j}). \tag{16}$$

ところで，式 (16) は，トークンと文を同一視する制約を加えているとき，式 (4) の第二項と一致する．すなわち，トークンスコア関数は，式 (4) の第二項を文圧縮に対応できるように拡張したものとみなすことができる．また，要約の対象となる文書が単一の文により構成される場合，式 (16) は式 (7) と一致する．すなわち，トークンスコア関数は，式 (7) を重要文抽出に対応できるように拡張したものとみなすこともできる．

6.2 単語スコア関数

単語に対してスコアを付与し，要約に含まれる単語のスコアの合計を要約のスコアとするスコア関数を，単語スコア関数と呼ぶ．単語スコア関数は以下のように定式化される:

$$\sum_{v \in V} \theta_v R(v). \tag{17}$$

このスコア関数を利用することにより，同じ単語が高々1 回しかスコアの対象にならなくなる．そのため，同じ単語が繰り返し要約に含まれることを是正し，重要な単語をより多く入れることができるようになる．スコアは tf · idf を用いる．すなわち，

$$R(v) = f_v \log \frac{F}{F_v} \tag{18}$$

である．

なお，式 (18) と式 (4) の第一項は，式としては同じだが R(v) の定義が異なることに注意する．

6.3 語と文の整合性制約

トークンと文の決定変数に矛盾が生じることのないようにするための制約を導入する．まず，トークン $w_{i,j}$ が要約に残るためには，文 s_i が要約に残っていることが必要である．そのため，$1 \leq i \leq m$, $1 \leq j \leq n_i$ に対し，制約

$$\delta_{i,j} \leq \xi_i \tag{19}$$

を追加する．

また，文 s_i が要約に残るためには，文 s_i 中の少なくとも一つのトークン $w_{i,j}$(ただし記号は除く) が要約に残ることが必要である．そのため，$1 \leq i \leq m$ に対し，制約

$$\sum_{j=1; w_{i,j} \notin MARK}^{n_i} \delta_{i,j} \geq \xi_i \tag{20}$$

を追加する．

6.4 トークンと文を同一視する制約

文圧縮を行わず重要文抽出を行うならば，トークン $w_{i,j}$ の決定変数 $\delta_{i,j}$ は文 s_i の決定変数 ξ_i と一致する．すなわち，$1 \leq i \leq m$, $1 \leq j \leq n_i$ に対し，制約

$$\delta_{i,j} = \xi_i \tag{21}$$

を追加する．

6.5 トークンと単語の整合性制約

トークンと単語の整合性を保つため，任意の単語 v に対し，制約

$$\sum_{w_{i,j}; w_{i,j}=v} \delta_{i,j} \geq \theta_v \tag{22}$$

を追加する．この制約は，同じ単語の全てのトークンが要約に含まれないならば，その単語は要約に含まれることを禁止することを表している．

なお, トークンと文を同一視する制約を加えると, 式 (22) は式 (5) と一致する. すなわち, トークンと単語の整合性制約は, 式 (5) を文圧縮に対応できるように拡張したものとみなすことができる.

6.6 構文制約

文法構造を維持しないまま要約を作成しようとすれば, より多くの情報を持つトークンを要約に含めることができるものの, 読みやすさは大きく低下する. 要約は人が読むものである以上, 読みやすさを無視することはできない. そのため, 一定以上の読みやすさを確保するために, 文法構造を制約として追加する必要がある. この制約を構文制約と呼ぶ. 構文制約は Clark ら (2008) の要約長制約を除く制約に修正を加えたものである.

なお, 本稿では英語を対象に制約を構成しているが, 他の言語でも同様の制約を構成することができれば, 提案手法を利用することができるだろう.

修飾子制約

Clark らの同名の制約と同一である.

文型制約

Clark らの同名の制約に加え, 動詞と補語の関係にあるトークンに対しても制約を持たせる. すなわち, トークン $w_{i,j}$ が, 動詞 $w_{i,k}$ の主語, 目的語, 補語 (係り受け関係:NCSUBJ, XSUBJ, CSUBJ, DOBJ, OBJ2, OBJ, IOBJ, XCOMP, CCOMP, PCOMP) であるとき, 制約

$$\delta_{i,j} = \delta_{i,k} \tag{23}$$

を追加する.

助動詞制約

助動詞は本動詞の時制や態を変化させるために使われるものであり, 助動詞が要約に含まれるか否かで本動詞の態や相などが変化してしまう. そのため, これらのトークンに対し制約を持たせる. すなわち, 動詞 $w_{i,j}$ が, 助動詞 $w_{i,k}$ の本動詞である (係り受け関係:AUX) とき, 制約

$$\delta_{i,j} = \delta_{i,k} \tag{24}$$

を追加する.

節句制約

Clark らの同名の制約で対象になっている節および句に加え, カンマから始まる節 (品詞:*comma*), 会話文 (品詞:*quo*), to 不定詞 (品詞:V1/to_bse/-) も対象にする.

that 節制約

RASP において, 一部の that 節は品詞が割り当てられない. 代わりに係り受け関係として現れる. 全ての that 節に対し適切な制約を付与するために, この RASP 特有な関係にも制約を持たせる必要がある. すなわち, トークン $w_{i,k}$ とトークン $w_{i,l}$ が $w_{i,j}$ ='that' による係り受け関係 (サブタイプ:that) を持つとき, 制約

$$\delta_{i,j} = \delta_{i,k} = \delta_{i,l} \tag{25}$$

を加える.

等位接続詞制約

先ほども述べた通り, Clark らの同名の制約には不備があるため, 等位接続詞に関する制約を新たに提案する. 等位接続詞で結ばれた二つのトークンは, 同程度の重要性を持っていると考えられる. そのため, 等位接続詞のトークンを含めた 3 つのトークンを一つの抽出単位として扱う. すなわち, トークン $w_{i,k}$ とトークン $w_{i,l}$ が等位接続詞 $w_{i,j}$ で接続されている (CONJ) とき, 制約

$$\delta_{i,j} = \delta_{i,k} = \delta_{i,l} \tag{26}$$

を加える.

アポストロフィ制約

RASP はアポストロフィを持つトークンを 2 つのトークンに分離する. この 2 つのトークンを一つの抽出単位として扱う. トークン $w_{i,j-1}$ とアポストロフィを含むトークン $w_{i,j}$ が文書中に存在するとき, 制約

$$\delta_{i,j-1} = \delta_{i,j} \tag{27}$$

を加える.

構文解析誤り補正制約

構文制約の対象にならなかったトークンは、他のトークンとは独立に要約に含むことができてしまう. このようなトークンが生まれてしまう原因は係り受け解析誤りである. そこで、このようなトークンは要約の対象に含めないこととする. すなわち, トークン $w_{i,j}$ が, この制約を除く構文制約の対象にならなかったとき, 以下の制約を加える:

$$\delta_{i,j} = 0. \tag{28}$$

6.7 要約長制約

要約長の上限を設ける. K 語以内の要約を作成するとき, 以下の制約を追加する:

$$\sum_{i=1}^{m} \sum_{j=1}^{n_i} \delta_{i,j} \leq K \tag{29}$$

以上のスコア関数と制約を組み合わせることにより, モデルを作る. なお, 先ほども述べたように, スコア関数は線形補間する. この時の補間係数を λ とする. すなわち, スコア関数は,

$$(1-\lambda)\sum_{i=1}^{m}\sum_{j=1}^{n_i}\delta_{i,j}I(w_{i,j})+\lambda\sum_{v\in V}\theta_v R(v) \quad (30)$$

となる. このスコア関数に制約を加えることにより, 自動要約を整数計画問題として定式化する.

7 実験設定

提案手法を評価するために, 英語を対象にした複数文書の 100 語要約を行う.

まず, 実験のために使用したツールを紹介する. idf の計算のための unigram 生成に, CMU-Cambridge Language Modeling Toolkit(Clarkson(1997)) を用いた. 構文制約生成のための係り受け解析には RASP(Briscoe and Carroll, 2002) を用いた. デコーディングには CPLEX(1998) を用いた. 前処理として, DUC2003(2003) で提供された文境界検出スクリプトを用いて, 文境界を検出している.

次に, コーパスを紹介する. idf の計算のために reuter コーパスを用いた. モデルの結合における補間係数 λ の学習のためのデータとして DUC2003 複数文書要約タスクの task2 のデータセットを用いた. 学習方法として, 0.1 刻みで λ の値を変えていき, ROUGE-1 の値が最も大きかった λ を用いることにした. テストデータに DUC2004 複数文書要約タスク (2004) の task2 のデータセットを用いた.

最後に, 評価方法を紹介する. 評価方法は自動評価システム ROUGE(Lin, 2004) を用いる. ROUGE はニュースドメインにおいて人手で行った評価と高い相関を持つことが知られている. ROUGE-1, ROUGE-2, ROUGE-SU4 に対し実験を行った. ストップワードは ROUGE のストップワードを用いている[1]. 提案手法は comp-select と表記する. ベースラインとして greedyline と stack-train-sim, それに, 本稿で紹介している重要文抽出のみ (comp) を行うモデルとする. greedyline は文書クラスタの中で最も新しい文書の先頭 100 語を要約とする手法である. greedyline は DUC2004 複数文書要約タスクでもベースラインとして用いられている. また, stack-train-sim は Yih らが提案している複数の手法の中で ROUGE-1 の値が最も高かったシステムである.

[1]利用したオプションは以下のとおりである.
ROUGE-1.5.5.pl -n 3 -x -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d -s -m -l 100

8 評価

実験結果を表 1 に示す. 全てのシステムの中で最も良い値を斜体で記述しており, その値より統計的に有意な差 (95%) がない値をボールド体で記述している. 提案手法である comp-select が ROUGE-1 において stack-train-sim を大きく上回った. しかし, ROUGE-2, ROUGE-SU4 において, stack-train-sim を下回っている. ただし, ROUGE-2 に関しては統計的有意な差があるとは言えない. greedyline との比較では, 全ての評価において大きく上回った. comp との比較では, ROUGE-1, ROUGE-SU4 については上回ったものの, ROUGE-2 は低下した. 全体の傾向として, ROUGE-1 は劇的に改善され, ROUGE-2 が統計的に有意では無いものの低下した. 文圧縮を重要文抽出と同時に行うことにより, より重要な語が要約中に含まれるようになったが, 同時に読みやすさの低下が起こったものと考えられる.

次に, δ と ROUGE 値との関係を見てみよう. ディベロップメントデータを用いた学習における, λ と ROUGE-1, ROUGE-2, ROUGE-SU4 の値の推移を図 1, 2, 3 にそれぞれ示す. ROUGE-1 においては $\lambda = 0.9$ が最も良い数値を出している. 一方, ROUGE-2 においては $\lambda = 0.7$, ROUGE-SU4 においては $\lambda = 0.8$ が最も良いことが分かる. また, $\lambda = 0.0$, 即ちスコア関数がトークンスコア関数であるとき, 全ての評価において comp-select が comp を下回っている. 文書ごとに表現の重なりが存在すると考えられるため, 複数文書を対象とした要約では, 冗長性を減らすことが重要となる. しかし, トークンスコア関数は, 異なるトークンであれば同じ単語であってもスコアの対象になるため, 冗長性を減らすことが出来ず, ROUGE 値が低くなってしまったものと考えられる. ただし, 同じ単語を重複なしでスコア付けする単語スコア関数のみを適用するよりも, トークンスコア関数を結合するほうが結果が良い. よって, 複数のモデルから成るスコア関数を作成することが, 要約候補のスコア付け方法として適切であると考えられる.

最後に, 実際に出力された要約を見てみよう. システムが出力した要約の一例を図 4 に示す. なお, この例の文書は 208 文あった. 図 4 を見ると, ほとんどの文が正しい文法性を維持したまま圧縮されて出力されている. また, 208 文あった文が 14 文まで制限されて出力されている. 重要文抽出と文圧縮が共に機能していることが見て取れる. ただし, 一部の文において, 改善すべき項目が見て取れる. 例えば, 文法上は正しいが, 極端に短い為に十分な情報が含まれていない文がある (10 文目). また, 不要な読点が含まれている文が散見

表 1: 実験結果

	ROUGE-1	ROUGE-2	ROUGE-SU4
comp-select($\lambda = 0.9$)	***0.370***	**0.079**	0.115
comp($\lambda = 0.8$)	0.332	**0.083**	0.107
Baseline	0.202	0.061	0.099
stack-train-sim	0.339	***0.086***	***0.129***

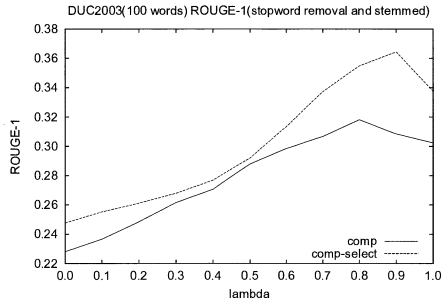

図 1: ROUGE-1

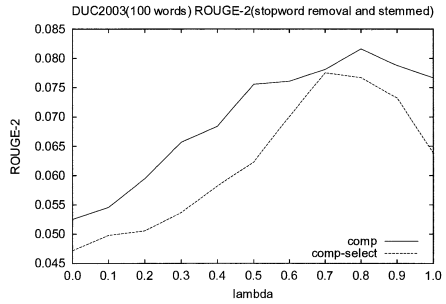

図 2: ROUGE-2

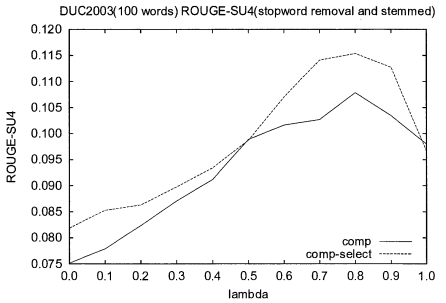

図 3: ROUGE-SU4

される. さらに, 非文法的な文もある (2 文目). これは構文解析器による解析誤りによるものと考えられ, 完全に回避するのは難しい. このような要約に残すべきでない抽出単位を, スコア関数や制約, あるいは前処理, 後処理で取り除くことにより, 性能のさらなる向上が期待できる.

9 結論

整数計画法を用いて, 新しい抽出的要約システムを構成した. この手法は重要文抽出と文圧縮を同時に行うシステムである. tf·idf に基づく単純なスコアであるにも関わらず、過去のシステムを ROUGE-1 において大きく上回った.

今後の発展について, 短期的には, 読みやすさに関する詳しい調査を行いたい. 他にも 8 節で示したとおり, 構文制約の改良と不要な記号の除去を行いたいと考えている.

長期的には 2 つの方向性を考えている. 第一の方向性は性能の改善である. 今回提案したモデルは, 他のスコア関数を利用することができる. さらに, 今回は提案していないが, 文のスコア関数を追加することも考えられる. より多くのスコア関数を追加することで, よりよい要約を作ることができるだろう. しかしながら, スコア関数が多くなってくると, 全ての補間係数の値の組み合わせを探すのが困難になってくる. そのため, より効率的に補間係数を探すことができるような, 学習手法の導入が求められる.

第二の方向性は他のドメインへの適用である. 本稿では, DUC の文書を対象にしてきたため, ニュースドメインにおける自動要約を行ってきた. しかし, 自動要約への要求はニュースのみに限らず, 他のドメインに適用することを考えたい. ただし, 他のドメインにおける自動要約の状況はあまり芳しくない. たとえば, ニュースドメインにおいては ROUGE が人手による評価と相関があることが Lin(2004) にて示されているが, 他のドメインでは, 会議ドメインにおける ROUGE と人手による評価との相関について調査した Liu ら (2008) を除いては示されていない. このようなニュースドメイン以外のドメインの状況も考慮し

1: Qin, detained, was questioned.
2: , not Beijing police.
3: Wang was student leader in influential Tiananmen Square movement.
4: leading advocate of China Democracy Party, Wang Youcai, was Hong Kong-based Information Center of Human Rights and Democratic Movement reported.
5: It was international human rights treaty China has signed.
6: member of dissident community, Xu Wenli, was arrested, charged.
7: dissidents signed letter to Chinese government protesting detentions,.
8: Xu spent years in prison.
9: Washington gave Yao visa,.
10: family had life,.
11: defendant is Yongmin political party.
12: He has defense lawyer.
13: trials of prominent democracy advocates for inciting subversion state opened Thursday morning,.
14: said Wang's wife, Hu Jiangxia,.

図 4: DUC2004 d30022t 要約 (100 単語)

た上で, 他のドメインへの適用方法を慎重に考えたい.

参考文献

Ted Briscoe and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation(LREC)*, pages 1499–1504.

James Clark and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Philip Clarkson and Ronald Rosenfeld. 1997. Statistical language modeling using the CMU-Cambridge Toolkit. In *Proceedings of Eurospeech'97*, pages 2707–2710.

John M. Conroy, Judith D. Schlesinger, and Jade Goldstein. 2005. Three classy ways to perform Arabic and English multi-document summarization. In *Multi-Lingual Summarization Evaluation*.

DUC. 2003. Document Understanding Conference. In *HLT/NAACL Workshop on Text Summarization*.

DUC. 2004. Document Understanding Conference. In *HLT/NAACL Workshop on Text Summarization*.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 177–185, October.

Chiori Hori and Sadaoki Furui. 2004. Speech summarization: an approach through word extraction and a method for evaluation. *IEICE Transactions on Information and Systems*, E87-D(1):15–25.

ILOG. 1998. *ILOG CPLEX User's Manual*. Lucent Technologies.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceedings of the 17th National Conference on American Association for Artificial Intelligence*, pages 703–710.

Chin-Yew Lin. 2004. ROUGE: A package for Automatic Evaluation of Summaries. In *Workshop on Text Summarization Branches Out at ACL*, pages 1585–1588.

Feifan Liu and Yang Liu. 2008. Correlation between ROUGE and Human Evaluation of Extractive Meeting Summaries. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 201–204.

Wen-Tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. 2007. Multi-document summarization by maximizing informative content-words. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1776–1782.

高村 大也 , 奥村 学. 2008. 最大被覆問題とその変種による文書要約モデル. 人工知能学会論文誌, 23(6):505–513.