3B1-6

単語の概念関係による文間の一貫性を用いた文圧縮システム

A sentence compression system considering inter-sentential consistencies based on semantic relations between words

橋本 哲弥[*1] 鶴岡 慶雅[*2] 近山 隆[*2]
Tetsuya Hashimoto Yoshimasa Tsuruoka Takashi Chikayama

東京大学大学院 情報理工学系研究科[*1]
Graduate School of information Science and Technology, The University of Tokyo

東京大学大学院 工学系研究科[*2]
Graduate School of Engineering, The University of Tokyo

Consistency of sentences is an important factor in text summarization. In our method, we use the evaluation method of sentences consistency with words semantic relation for sentence compression to make compressed sentences more fluent. The results show that our method's F-score of grammatical relations fell, but our method achieves improvement of precision of Simple String Accuracy.

1. 背景と目的

近年，コンピュータや携帯電話などのメディアの普及により，インターネット上における電子テキストデータの量は増え続けている．しかし, 一方で人間の情報処理能力には限界がある．そのため, 各々のユーザーが求める情報をより効率的に抽出する技術の必要性が高まっている．そのような技術の代表的なものとして，検索エンジンが挙げられる．検索エンジンでは，与えられたクエリから，それに関連するページを絞り込み，関連度が高い順に並べてユーザーに提示する．しかし，与えられたクエリが同じであってもユーザーによって求めている情報が異なることがあるため，最終的にはユーザーが検索エンジンによって提示されたページを確認し，求めている情報を探す必要がある．このような場合，候補ページと共にページ内の文章の要約を表示することで，ユーザーの負担を軽減することができる．自動要約技術にはこの他にも，携帯電話やスマートフォンなどの画面のリソースが限られているハードウェアのために文を出来る限り短くするなどの利用法も存在する．

既存の自動要約システムの多くは，文章中の単語の出現頻度から計算した各単語の重要度を用いて，重要文を抽出し，その後処理，またはその前処理として各文中に存在する冗長的な表現を取り除く文圧縮を行っている [McDonald 07]．文圧縮では，入力として与えられた文から構文木を生成し，各句・単語が削除可能かどうかを学習データに基づいて判断するという手法が一般的であり, 削除規則の学習には原文と人手による圧縮文が組となっているデータを使用する．しかし，この圧縮文からの学習では文章中の文同士の持つ関係を考慮せず，文法的に削除しても問題ないかどうかだけを判断しているために, 文法的には正しいが他の文との間で不自然な圧縮を行なってしまうことがある．故に, より精確な文圧縮を行うには文法的な正しさに加えて文章の構成を考慮する必要がある．

これらの観点から， 本研究では， 文法的な情報のみではなく， 各文間の情報を使用することで， より自然かつ文意を保った文圧縮を行う自動要約システムを目的とする．

提案手法では, 文章中の全ての文に対する圧縮文候補のうち最も正しいと考えられる 1 文を圧縮文として選択し, 決定した圧縮文と一貫性の高い圧縮文候補を優先的に圧縮文として選択するように, 既に決定した圧縮文によって各圧縮文候補の一貫性評価を更新する．これによって文章全体の一貫性を考慮した文圧縮を実現する．

連絡先: 連絡先：橋本哲弥, 東京大学大学院情報理工学系研究科, hashib@logos.t.u-tokyo.ac.jp

2. 関連研究

2.1 2 段階の処理による文圧縮

これまでに高精度の結果を出している文圧縮システムとしては, Dimitrios ら [Dimitrios 10] による手法が挙げられる．Dimitrios らの手法では文圧縮を, 圧縮文候補の生成という第 1 段階と, 生成された圧縮文候補の文単位での再評価という第 2 段階に分割した 2 ステップの処理によって行なっている．また, 人手によって足されたルールが無く, 全ての処理を機械学習によって行なっているのも特徴の 1 つである．

第 1 段階の圧縮文候補の生成では, 原文と圧縮文の組からなるコーパスの原文と圧縮文それぞれから依存関係木を生成し, 関係木上の各エッジに対して,「そのエッジ以下の木を除去」,「そのエッジ以下の木を抽出」,「除去を行わない」の 3 種類の処理のうちどれを行っているかを学習する．学習には Maximum Entropy framework を用い, 周辺の特徴量から各エッジに対してどの処理を行うかをクラス分け問題として扱う．また, それぞれの処理が行われる確率をエッジ毎に計算し, それらを掛けあわせたものが閾値以下となる圧縮文候補は無視することで計算量を削減している．

第 2 段階では, 第 1 段階で生成された圧縮文候補に対して trigram による文の生成確率, 原文と圧縮文候補の間の tf-idf の比による重要度, 除去された単語と除去されなかった単語の依存関係木上での深さ, 原文に含まれていた各品詞の数とそれらの内圧縮文において除去されたものの数などといった, 第 1 段階の処理で扱えなかった圧縮文の文単位から得られる情報を特徴量として用いて, 各圧縮文候補のスコアを計算し, 再評価を行う．

2.2 文同士の概念相似度に基づく一貫性評価

WordNet[Fellbaum 05] の階層構造から計算される単語間概念相似度を文と文の間に適用することで, 段落の一貫性の評価指標とする手法が板倉ら [板倉 08] によって提案されている．この手法では, 段落を構成する各文に対して単語の概念相似度か

ら文と文の話題の関連度 R_i を計算し, その平均値を段落の一貫性の評価値として定義している. 段落に含まれる文の数を n, 段落の一貫性の評価値を C とすると, C は式 (1) によって計算される.

$$C = \frac{1}{n}\sum_i R_i \tag{1}$$

ここで, 各文の関連度 R は, 段落中のある文 S_i に出現する単語集合を $W(S_i) = \{w_{i1}, w_{i2}, ..., w_{im}\}$, S_i 以外の文で出現する単語集合を $W(P_{S_i}) = \{w_{p1}, w_{p2}, ...\}$ とすると, 単語間の概念相似度関数 $Sim(w_a, w_b)$ を用いて式 (2) のように計算される. また, 概念相似度は WordNet 上での階層構造において 2 つの単語間に存在する最短経路の長さから計算される.

$$R_i = \frac{1}{m}\sum_i \max_j Sim(w_i, w_{pj}) \tag{2}$$

式 (1) 及び式 (2) によって計算される段落一貫度 C は段落の内容一貫性評価尺度であり, 段落を構成する各文の含まれる単語の概念関係における関係性の強さを示している.

2.2.1 話題ネットワーク

渋木ら [Shibuki 05] は, 文章中の単語を入力とし, 入力された名詞と動詞をノードとしたネットワークの活性値に基づいて, 現在の主題を判断するという手法を提案している. この手法では, 各文を読んだ時点での話題の移り変わりを, ネットワーク上の活性値によって表現し, 全ての活性値を常に減少させることで話題の変化を, 入力された単語のノード及びそれに関するノードの活性値を増加させることで話題を留めようとする力をそれぞれモデル化する. これらの力を拮抗させることで, 現在の話題を求める.

3. 提案手法

本手法では板倉らの複数文間の一貫性評価尺度を文圧縮に用いることで, より自然な圧縮文の生成を目的とする. そこで, Dimitrios らの手法を基にした 3 段階の処理によって文圧縮システムを実装した. 第 1 段階では, 原文・圧縮文の組からなるコーパスより圧縮規則を条件付き確率場 (Conditional Random Fields, 以下 CRF) によって学習し圧縮文の候補を生成する. そして第 2 段階では, 条件付き確率場によって生成された圧縮文候補を, 文単位の評価によって Support Vector Regression (以下 SVR) によって再評価 (reranking) する. さらに, 第 2 段階によってスコアが最も高く順位付けされた文を圧縮文として決定することによって, まだ決定していない圧縮文候補の一貫性の評価値を更新し, その値を用いて周囲の文を第 2 段階の処理によって再評価するという 3 段階の処理によって文圧縮を行う.

3.1 CRF による単語単位の評価による圧縮文候補生成

教師データとして原文・圧縮文の組からなるコーパスを用い, CRF によって圧縮規則を学習する. これを基に与えられた文中の各単語について除去を行うか否かのラベル分類を行うことで, 圧縮文候補を生成する. CRF では, 入力記号列 x に対して出力系列 y を式 (3) のようにモデル化する. 本手法においては, x が単語列, y が各単語に対する除去を行うか行わないかのラベルとなる.

$$P(y \mid x) = \frac{1}{Z(x)}\exp(\sum_j \lambda_j F_j(x, y)) \tag{3}$$

ここで, $Z(x)$ は全系列を考慮したときに確率の和が 0 となるための正規化項を表す. また, $F(x, y)$ は出力系列 y 上の全てのパスについての素性ベクトルを足しあわせたものを表す.

CRF に与える特徴量としては以下を用いた.

1. 単語とその bigram, trigram
2. 品詞とその bigram, trigram
3. 依存関係木上での親ノードの品詞と依存関係
4. 依存関係木上での親ノードの親ノードの品詞と依存関係
5. 依存関係木上での子ノードとの品詞と依存関係
6. 2, 3, 4, 5 を組み合わせたもの
7. 依存関係木上での現ノードの深さ
8. tf-idf による単語の重要度

これらの特徴量と, 教師データにおけるラベルを基に, 入力文に対して各単語を除去するか否かのラベル付けを行い, CRF から出力される条件付き確率が一定値以上のものを圧縮文候補として出力する.

3.2 SVR による圧縮文候補の文単位での再評価

CRF によって生成された圧縮文候補を SVR を用いて文単位で再評価する. SVR では, 入力ベクトル x に対応する出力値 y を, 教師データとして与えられる (x, y) の組から学習する. 本手法では, 入力ベクトル x として第 1 段階で生成された圧縮文候補から得られる特徴量を与え, 出力 y としては Dimitrios らの研究を参考に式 (4) によってスコアを計算した.

$$\begin{aligned} y_i = (\lambda_1 * SSA(c_i \mid s_i) + (1 - \lambda_1) Gramm(c_i)) * \lambda_2 \\ + C(c_i) * (1 - \lambda_2) - \alpha\ CR(c_i) \end{aligned} \tag{4}$$

ここで λ_1, λ_2, α はパラメータ変数であり,c_i は原文 s_i に対する圧縮文候補, $Gramm(c_i)$ は n-gram から計算される圧縮文候補の生成確率, $C(c_i)$ は単語の概念関係度から計算される圧縮文候補と周囲の文との一貫性の評価指標である. また, $SSA(c_i \mid s_i)$ は出力データと正解データに対して挿入 (Insertion) , 削除 (Deletion) , 置換 (Substitution) の処理を行なって, 出力データと正解データが一致するまでに何回の処理が必要かを測ることで, 出力データと正解データの間に存在する誤差を計算する尺度であり, 式で表すと式 (5) のようになる.

$$SSA(c_i \mid s_i) = (1 - \frac{I + D + S}{R}) \tag{5}$$

ここで, I, D, S は出力データが正解データと一致するまでに必要となる挿入・削除・置換の処理の数であり, R は正解データの文中に存在する単語数を示している.

入力ベクトルが持つ特徴量としては Dimitrios らの手法を参考に, 以下のものを用いた.

- CRF による圧縮文候補の生成確率
- 単語の trigram による文の生成確率
- 品詞の trigram による文の生成確率
- 原文と圧縮文候補に含まれる単語の重要度の総和の比 (動詞・名詞のみ)
- 除去された語と除去されなかった語のそれぞれの依存関係木上での深さの平均
- 原文中に出現する各品詞の数と, それらのうち除去された各品詞の数
- 周囲の文 (原文) との関連度に文間距離の補正を掛けて足しあわせたもの

また, 出力 y の教師データとしてはトレーニングデータで学習を行った CRF によってトレーニングデータの圧縮文候補を生成したものに式 (6) によってスコアを計算したものを用いた.

3.3 圧縮文の決定と文間一貫性の特徴量の更新

SVR による出力 y の値が最も高い圧縮文候補をその基となる文の圧縮文として決定し, その周囲の文との一貫性の特徴量を更新する.

一貫性を表す特徴量は, 現在の文と周囲の文に含まれる動詞・名詞に対して関連度を計算することで求める. また, このとき文同士の文章中での距離を二乗したものを係数として用いることで, 近い文との一貫性を重視するようにする. 圧縮文候補 S_i の周囲の文との一貫性 $C(S_i)$ は,

$$C(S_i) = R(S_i, S_{i+1}) + \frac{1}{2^2}R(S_i, S_{i+2}) + \frac{1}{3^2}R(S_i, S_{i+3}) + ... \\ +R(S_i, S_{i-1}) + \frac{1}{2^2}R(S_i, S_{i-2}) + \frac{1}{3^2}R(S_i, S_{i-3})...(6)$$

と表される. 実際の処理の際には話題ネットワークの考え方を基として, より局所的な一貫性評価を行うために文間距離が一定値以上離れた場合, その関連度 R を無視するようにした.

この「SVR による圧縮文候補の文単位での再評価」という 2 段階めの処理と,「出力 y が最大となるものを圧縮文として決定し, その周囲の文との一貫性を更新する」という 3 段階めの処理を文章中の全ての文の圧縮文が決定されるまで繰り返すことで文圧縮を行う.

4. 実験

原文と人手による圧縮文の組からなるコーパスを用い, 人手による圧縮文を正解データとすることでシステムの圧縮文を評価する. 評価, 及び学習に用いるコーパスとしては Edinburgh の Written News Compression Corpus (WNC Corpus)[James 00] を使用した. WNC Corpus はニュースの書き下しテキストからなるコーパスであり, ニュースの原文と, その原文から単語の削除のみによって生成された圧縮文が組となって与えられている. これを性能の比較のために, Dimitrios らと同じようにトレーニングデータが 1024 組, 開発データが 324 組, テストデータが 291 組となるように分割し, 開発, 学習, 及び評価に使用した. また, WNC Corpus は圧縮文と原文の間にアラインメントが付けられていないため, SSA の計算に必要な原文と圧縮文の間の各単語の対応付けを Smith-Waterman アルゴリズム [Smith 81] によって行った. また, 既存のライブラリとして圧縮文候補の生成には CRF++[Taku 05], 文単位での再評価には libsvm[Chang 01] をそれぞれ使用した.

圧縮文の評価尺度としては, 前述の SSA による評価に加えて RASP[Briscoe 02] を用いた GR (Grammatical Relation) の F1-measure を用いた. GR の F1-measure では, 正解データと出力データのそれぞれに対して RASP によって依存関係木を生成し, 正解データ中に現れる単語の依存関係が出力データ中においてどの程度現れているかを適合率と再現率, 及びその調和平均である F 値によって評価する. GR を用いた評価は人手による評価と高い相関があるとされており, 多くの文圧縮手法においてこの指標による評価が行われている [James 06].

ベースラインとしては, 2 段階目の処理において出力 y の計算に式 (7)（式 (4) から文間一貫性の項を取り除いたもの）を用い, 3 段階目の処理を行わずに各文において出力 y が最大となったものを圧縮文としたものを用いる.

$$y_i = (\lambda_1 * SSA(c_i \mid s_i) + (1 - \lambda_1)Gramm(c_i))$$

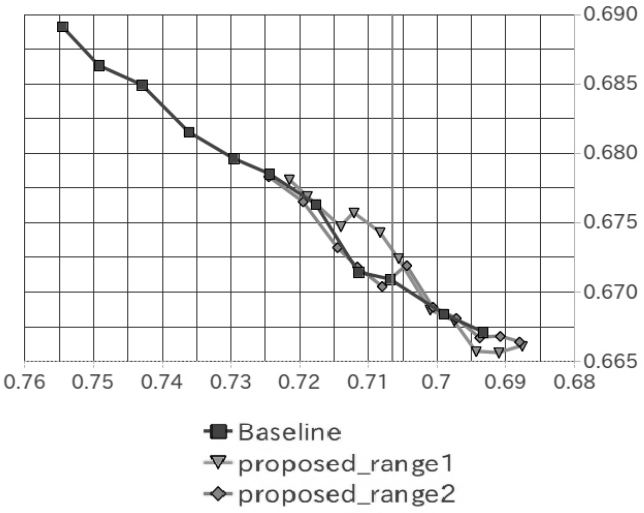

図 1　各圧縮率に対する SSA による評価

$$- \alpha\ CR(c_i) \quad (7)$$

また, 提案手法においては各文に対して前後一文との一貫性を計算したものと, 前後二文に対して一貫性を計算したものをそれぞれ実験に使用した.

5. 評価

開発データにおいて出力 y の計算式の圧縮率に関するパラメータ変数 α を 0.00 から 0.10 まで 0.01 刻みで変化させ, 横軸に圧縮率, 縦軸に SSA をとったものを図 1 に, 横軸に圧縮率, 縦軸に F1-measure による評価をとったものを図 2 にそれぞれ示す. ここで, proposed_ range1, proposed_ range2 はそれぞれ提案手法において各文に対して前後一文との一貫性を計算したものと, 前後二文に対して一貫性を計算したものを表す. また, 正解データ全体の圧縮率 0.7072 を図 1, 図 2 中に示した.

図 1, 2 の結果から, 提案手法ではベースラインに比べて圧縮率が全体的に高くなっていることが分かる. また, SSA による評価では正解データの圧縮率付近において提案手法の精度がベースラインの精度を上回っていることが分かる. その一方で図 2 の結果では, 全体を通して提案手法の精度がベースラインの精度を下回っている様子が見られる. SSA による評価は, 正解データとシステムの出力の単純な一致度を表しているため, 提案手法はベースラインに比べてより正解データに近い圧縮を行っているといえる. しかし, F1-measure による評価ではベースラインの精度を下回っていることから, 提案手法による圧縮では, 正解データ中に出現する単語の依存関係が保たれていないことが多いといえる. これらのことから, 提案手法では圧縮可能かどうかの判定の精度は上がっているが, その結果依存関係を考慮していない圧縮を行ってしまったことにより原文の持つ文の依存関係を崩してしまっていると考えられる.

次に, テストデータに対して文圧縮を行った結果を表 1 に示す. ここで, 圧縮率のパラメータ変数 α には開発データにおいて最も正解データの圧縮率に近い値をとったものをそれぞれ用いた. また, gold data は正解データの値を表す.

これに対して, Dimitrios らの手法の結果では $CR = 0.6372$, $F1 = 53.75$ であった. これらを比較すると, Dimitrios らの結果の方が圧縮率が高いにも関わらず F1-measure による評価が高く, 提案手法及びベースラインが精度で劣っていることが

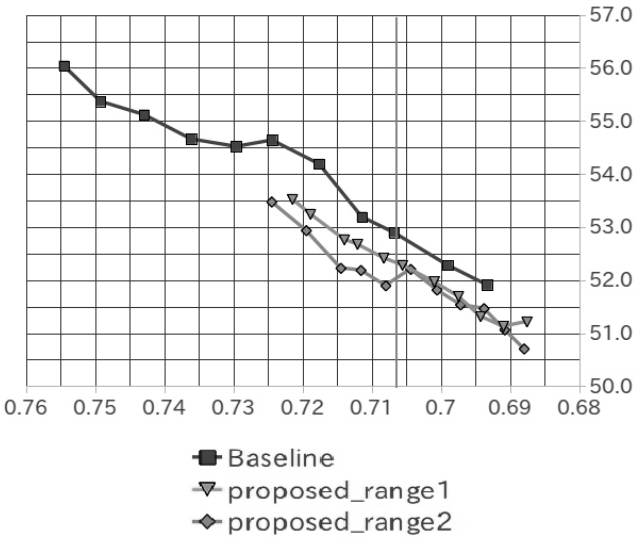

図 2　各圧縮率に対する F1-measure による評価

表 1　テストデータにおける各文圧縮手法の評価

system	SSA	CR	F1
Baseline	0.6710	0.7188	51.86
proposed_range1	0.6616	0.7073	50.79
proposed_range2	0.6590	0.6942	49.89
gold data	-	0.7043	100.0

分かる．ベースラインに用いた手法では, SVR による評価は Dimitrios らの手法と同じ特徴量を用いているため, 提案手法と Dimitrios らの手法の精度の差は 1 段階目の圧縮文候補の生成において生じていると考えられる．提案手法では文を入力列として, 各単語に関して除去するか否かを CRF によって特徴量から計算した．一方, Dimtrios らの手法では圧縮文候補の生成を依存関係木上の部分木の抽出及び除去によって行っている．このことから, 部分木の抽出・除去による圧縮文の生成では, 原文における依存関係が維持されるため, CRF による圧縮文候補を用いた提案手法よりも F1-measure による評価が高くなったと考えられる．

6.　まとめ

本稿では, 単語の概念関係度に基づいた文間の一貫性評価指標を用い, 1 つの圧縮文が決定する度に周囲の文との一貫性の評価を更新することで, 文同士の繋がりを重視した文圧縮手法を提案した．

結果, 人手による正解データに対して単語ごとの一致度ではベースラインに対して精度の上昇が見られた．しかし, 依存関係の F1-measure による評価ではベースラインにたいして精度の減少が見られた．これは, 圧縮文候補の生成において単語毎に除去を行うかを判定したために, もとの文の持つ依存関係を崩してしまっていることが原因であると考えられる．

参考文献

[McDonald 07] McDonald, R. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. Proceedings of 29th European Conference on IR Research, ECIR 2007, 557-564.

[Dimitrios 10] Dimitrios, G, and Ion, A. 2010. An extractive supervised two-stage method for sentence compression. Association for Computatinal Linguistics, 885-893

[Fellbaum 05] Fellbaum, C. 2005. WordNet and wordnets. In: Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 665-670

[板倉 08] 板倉 由知, 白井 治彦, 黒岩 丈介, 小高 知宏, 小倉 久和. 2008. 単語の概念関係を用いた段落の一貫性評価手法. 電子情報通信学会論文誌. D, 情報・システム J91-D(6), 1672-1675

[Shibuki 05] Shibuki, H., Araki, K., Momouchi, Y., and Tochinai K. 2005. A Proposition of a Method for Deep Case Analysis Based on Activity in Network, The Institute of Electronics Information and Communication Engineers, NLC, 105(204), 13-18

[Smith 81] Smith, T, and Waterman, M. 1981. “ Identification of common molecular subsequences, ” Journal of Molecular Biology,147(1): pp. 195-197,

[Briscoe 02] Briscoe, E. J. and J. Carroll. 2002. Robust accurate statistical annotation of general text. In Proceedings of the 3rd LREC. Las Palmas, Spain, 1499-1504.

[James 06] James, C. and Mirella, L. 2006. Models for sentence compression: a comparison across domains, training requirements and evaluation measures. ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, 377-384

[James 00] James, C. 2000. Written News Compression Corpus http://jamesclarke.net/research/resources

[Taku 05] Taku, K. 2005. CRF++: Yet another CRF toolkit

[Chang 01] Chang, C. C. and Lin, C. J. 2001. LIBSVM: a library for support vector machines, Software http://www.csie.ntu.edu.tw/ cjlin/libsvm/