

過去情報の内容選択を取り入れた スポーツダイジェストの自動生成

加藤 舜介
東京工業大学 情報理工学院
kato.s.aw@m.titech.ac.jp

徳永 健伸
東京工業大学 情報理工学院
take@c.titech.ac.jp

1 はじめに

スポーツダイジェストは、スポーツの試合が行われた後に、記者によって書かれる試合の総評である。スポーツダイジェストには、試合に出場したチーム・選手の活躍や直近の調子の良し悪し、試合の勝敗結果によるシーズン成績の変動などの記述が含まれており、実際に試合を見ることができなくても、それを読むと試合の結果や展開がわかるように書かれている。またスポーツの試合には、様々なチームや選手のプレイデータが記録される。プレイデータに含まれる内容は、例えばサッカーやバスケットボールでは得点数やアシスト数、野球では安打数や投球数などが挙げられる。本研究は、このようなプレイデータからスポーツダイジェストを自動で生成することを目的とする。

既存のダイジェスト生成モデルでは、入力データの内容選択を行う層を組み込んでいるが、そのほとんどが当日の試合情報のみから内容選択をおこなうため、連勝や連敗、直近の試合の平均記録などの過去の記録を踏まえた出力ができない。そこで本研究は、過去の試合情報も含めて内容選択をおこなうことにより、チームや選手の過去の記録を踏まえたダイジェストが生成可能なモデルを提案する。また、内容選択と文章生成を別々に学習することによって内容選択をより重視したモデルの学習方法を提案する。

本研究の貢献は以下の2点である。

- 内容選択層を取り入れた Transformer ベースの既存モデルに対し、損失関数の重みづけや学習済みエンコーダのファインチューニングによるデコーダの学習をおこなうことで、既存のモデルより精度が向上することを確認した。
- 過去情報を含むデータセットを作成し、既存のモデルに過去情報の内容選択層を取り入れたモ

	文数	割合 (%)
過去情報が不要	1,193	87.5
選手の過去情報が必要	92	6.7
チームの過去情報が必要	55	4.0
スポーツ史の過去情報が必要	27	1.8

表1 ダイジェストを構成する文の種類

デルを学習することで、過去情報も含めたダイジェストを生成できることを確認した。

2 関連研究

Wiseman らは、Data-to-Text タスクのためのデータとして ROTOWIRE を作成した [1]。このデータセットは、NBA の試合におけるチーム・選手のプレイデータと、記者によって書かれたダイジェストの組が集められたものであり、近年のスポーツダイジェストにおける Data-to-Text の研究で広く使われている。また、Data-to-Text タスクのための自動評価指標も提案している。ROTOWIRE を用いた研究として、Puduppully らは、入力データの内容選択をおこなない、どのデータをどの順序で出力するかを計画を学習するエンコーダ・デコーダモデルを提案した [2]。Iso らは、重要な入力データを選択・追跡して文章生成に用いるという、人間のダイジェストを書く過程を模倣したモデルを提案した [3]。Gong らは、Transformer モデルのエンコーダに入力データの内容選択をおこなう層を追加したモデルを提案した [4]。

ROTOWIRE からランダムに 100 試合分のダイジェストを抽出し、それらを構成する文の種類を調べたところ、表 1 のようになった。過去情報が必要な文には、直近の選手の平均得点の記録や、チームの連勝記録が記述されている文などがある。既存のモデルは、入力に過去情報を用いないため、全体の 12.5% の文を生成できない。本研究は、入力にチームの過去情報を加えることで、チームの直近の勝敗記録に関する出力ができるモデルを目指す。

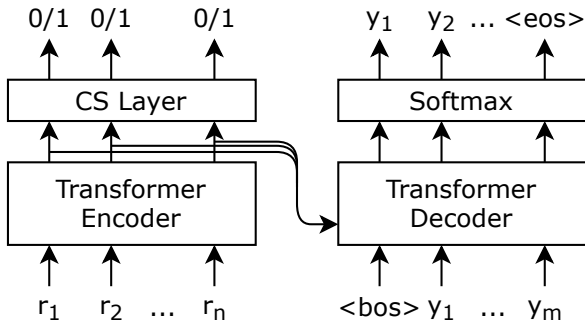


図1 Gongらのモデル [4]

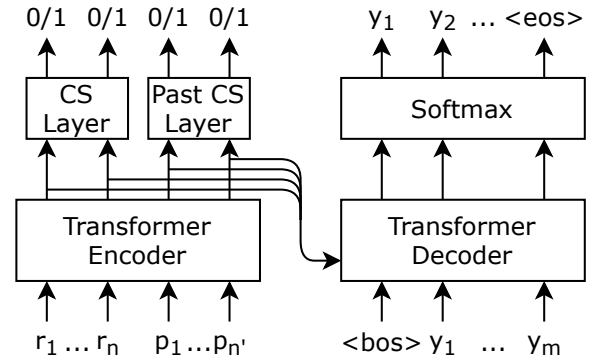


図2 提案モデル

3 提案手法

3.1 先行研究モデルの学習方法の変更

Gong らは、従来の Transformer モデル [5] に入力データの内容選択をおこなう層 (CS Layer) を追加したモデルを提案した (図 1). ただし、入力 r_1, r_2, \dots, r_n はトークン列に変換されたプレイデータである. これらの入力エンコーダでプレイデータベクトルに変換される. プレイデータベクトルは、デコーダの入力として使われることに加え、内容選択層の入力としても使われる. 内容選択層では、あるプレイデータをダイジェストに出力するかしないかの 2 値分類をおこなう. これにより、モデルは入力に含まれる複数のプレイデータの中から重要なデータを選択し、ダイジェストに出力する. このモデルのパラメタ θ_m は式 (1) のように定義される.

$$\theta_m = \arg \min(L_{cs} + L_{sg}) \quad (1)$$

ただし、 L_{cs} は内容選択の損失関数、 L_{sg} は文章生成の損失関数である. 本研究は、この損失関数に変更を加えた次の 2 つの学習方法を提案する.

- WT モデル：内容選択・文章生成の損失関数に重みづけをおこなう.

$$\theta_m = \arg \min((1-w) * L_{cs} + w * L_{sg}) \quad (2)$$

ただし w はハイパーパラメタである. これにより、内容選択をどの程度重視するかによる、モデルの出力への影響を調べることができる.

- FT モデル：内容選択層の学習を事前におこなったエンコーダを用いて、ファインチューニングによるデコーダの学習をおこなう.

$$\theta_e = \arg \min(L_{cs}) \quad (3)$$

$$\theta_m = \arg \min((1-w) * L_{cs} + w * L_{sg}) \quad (4)$$

ただし w はハイパーパラメタ、 θ_e はエンコーダのパラメタである. 内容選択の損失関数 L_{cs} を最小化する θ_e を初期値として (式 (3)), 重みづけした損失関数 $(1-w) * L_{cs} + w * L_{sg}$ を最小化する θ_m を学習する (式 (4)). これにより、エンコーダを独立して学習することで、内容選択が正確に学習されることが期待できる.

3.2 過去情報内容選択層の追加

Gong らのモデルは、入力に試合当日のプレイデータしか用いないため、過去のプレイデータを踏まえたダイジェストの出力ができない. 本研究は、入力にチームの過去の勝敗記録を加えることで、チームの連勝や連敗、直近の勝敗記録などの出力ができるモデルを提案する.

- PAST-int モデル：1 つの内容選択層で当日・過去の入力データから同時に内容選択をおこなう. 式 (2) を用いたモデル (PAST-int-WT) と式 (3), 式 (4) を用いたモデル (PAST-int-FT) を学習する.
- PAST-sep モデル：過去情報内容選択層 (Past CS Layer) を追加し、当日情報を CS Layer, 過去情報を Past CS Layer で内容選択をおこなう (図 2). ただし p_1, \dots, p'_n は過去情報の入力である. パラメタの学習は式 (5), 式 (6), 式 (7)+(8) の 3 つを考える.

$$\theta_m = \arg \min((1-w) * ((1-w_p) * L_{csn} + w_p * L_{csp}) + w * L_{sg}) \quad (5)$$

$$\theta_m = \arg \min((1-w) * L_{cs} + w * L_{sg}) \quad (6)$$

$$\theta_e = \arg \min(L_{cs}) \quad (7)$$

$$\theta_m = \arg \min((1-w) * L_{cs} + w * L_{sg}) \quad (8)$$

ただし w, w_p はハイパーパラメタ、 θ_m はモデル

全体のパラメタ, θ_e はエンコーダのパラメタであり, L_{csn} は図 2 における CS Layer の損失関数, L_{csp} は Past CS Layer の損失関数, L_{cs} は CS Layer と Past CS Layer の損失を連結した損失関数である. 式 (5) は CS Layer, Past CS Layer の内容選択の重みづけをおこなうモデル (PAST-sep-WT) である. 式 (6) は CS Layer と PAST CS Layer の損失を連結したモデル (PAST-sep-CONCAT) であり, PAST-sep-WT における重みを自動で学習することが期待できる. 式 (7)+(8) はファインチューニングモデル (PAST-sep-FT) であり, 考え方は FT と同じである.

4 評価実験

4.1 データセット

評価実験には Wiseman らの ROTOWIRE [1] を使用した. このデータセットの学習・検証・テストデータ数は, それぞれ 3,398, 727, 728 である. 入力のプレイデータに対する内容選択のラベルは Gong らと同じものを使用した.

元の ROTOWIRE はデータセットが時系列で分割されておらず, また同じプレイデータに対して 2 つのダイジェストが割り当てられている場合がある. そこで過去情報を用いたモデルには, ROTOWIRE から重複しているデータを削除し, 全データを時系列でソートして再分割したデータセット (ROTOWIRE-re) を使用した. これにより学習・検証・テストデータ数はそれぞれ 3,205, 309, 308 となった. 入力には試合当日のプレイデータに加え, 直近 5 試合分のチームの勝敗記録を過去情報として使用した. 過去情報に対する内容選択のラベルづけは, 100 試合分のダイジェストを調べたことにより作成した以下のルールでおこなった.

- 直近で 3 連勝 (3 連敗) 以上していた場合, それらの勝敗記録を選択する.
例: 過去 5 試合の結果が勝・勝・勝・負・負の場合, 勝を選択し, 負を選択しない.
- 直近 5 試合中 4 試合以上勝ち (負け) の場合, それらの勝敗記録を選択する.
例: 過去 5 試合の結果が負・負・勝・負・負の場合, 負を選択し, 勝を選択しない.

4.2 評価指標

モデルの評価には, 出力文の評価をおこなう BLEU [6] に加え, Wiseman らの提案した 3 つの評価指標を用いた. この指標は, 出力文が入力データの内容を正しく反映できているかを評価する.

- Relation Generation (RG): 出力文から (entity, value) の関係を抽出し, 抽出された関係の数と, それらの関係が入力データに対して正しいかどうかを評価する (Precision). ただし entity はチーム名や選手名などの動作の主体, value は得点数やアシスト数などの記録である.
- Content Selection (CS): 出力文とリファレンスから (entity, value) の関係を抽出し, 出力文から抽出された関係のリファレンスから抽出された関係に対する Precision, Recall で評価する.
- Content Ordering (CO): 出力文とリファレンスから (entity, value) の関係を抽出し, それらの間の正規化 Damerau-Levenshtein 距離 [7] で評価する.

これらの指標は当日情報に関する評価をおこなうため, 過去情報に関する評価はできない. そこで過去情報を用いたモデルについては, 自動評価に加えて以下の手順で人手評価をおこなった.

1. 学習データからランダムに 100 試合分のダイジェストを抽出し, 過去情報を含む文集合 P^+ と含まない文集合 P^- に分ける.
2. P^+ または P^- に含まれる単語 v に対してスコア $p(v)$ を以下の式で与える.

$$p(v) = \frac{C^+(v)}{|P^+|} - \frac{C^-(v)}{|P^-|}$$

ここで, $C^+(v), C^-(v)$ はそれぞれ P^+, P^- 中の v の頻度を表わす.

3. 各モデルの出力について出力中単語のスコア $p(\cdot)$ の総和を計算し, スコアが上位 30 件の出力について過去情報に言及している部分が正しいかどうかを手で確認し, 正解率を計算する.

4.3 モデルの詳細

全てのモデルは Gong らと同じパラメタ数を持つ Transformer ベースのモデルである. WT, FT の損失関数のハイパーパラメタは $w = 0.25, 0.5, 0.75$ とし, PAST-int, PAST-sep の損失関数のハイパーパラメタは $w = 0.25, w_p = 0.25, 0.5, 0.75$ とした. 各モデルは

表2 学習方法を変更したモデルの自動評価

モデル	RG (P% / #)	CS (P% / R%)	CO	BLEU
リファレンス	94.69 / 24.01	100 / 100	100	100
Gong ら	67.50 / 6.64	42.73 / 16.45	16.07	15.60
WT($w = 0.25$)	70.31 / 6.93	43.93 / 16.89	16.52	15.61
WT($w = 0.75$)	58.45 / 5.74	33.69 / 12.98	12.79	14.98
FT($w = 0.25$)	71.27 / 6.72	46.26 / 17.02	16.85	15.26
FT($w = 0.5$)	75.37 / 7.30	48.49 / 18.39	18.12	16.18
FT($w = 0.75$)	67.39 / 6.97	42.69 / 17.20	16.64	16.06

表3 既存モデルと FT の内容選択層の評価

モデル	Gong ら	FT
Precision(%)	73.18	67.35
Recall(%)	35.62	37.88
選択したプレイデータを 出力した数 (個 / 試合)	4.54	4.84
選択したプレイデータを 出力しなかった数 (個 / 試合)	11.30	14.91
選択していないプレイデータを 出力した数 (個 / 試合)	1.48	2.00
入力にないプレイデータを 出力した数 (個 / 試合)	3.61	2.94

10 回学習をおこない、その平均スコアを計算した。

4.4 実験結果

表2にWTとFTの結果を示す。WTにおいて $w = 0.5$ としたものがGongらのモデルに相当するが、 $w = 0.25$ とすると若干の性能向上が見られる。これらの中ではFT($w = 0.5$)のスコアが一番高い。

GongらのモデルとFT($w = 0.5$)における内容選択層の評価結果を表3に示す。FTの内容選択層のRecallが高く、選択したプレイデータを出力した数もGongらのモデルに比べて多い。FTの方が内容選択層で多くのプレイデータを選択したことで、入力にないプレイデータを出力することが少なくなり、スコアが高くなったと考えられる。

表4に過去情報を追加したモデルの実験結果を示す。比較のために、過去情報を追加していない ROTOWIRE-re で学習したGongらのモデルの結

表4 過去情報を追加したモデルの自動評価

モデル	RG (P% / #)	CS (P% / R%)	CO	BLEU
Gong ら	83.79 / 7.12	34.73 / 36.41	12.20	11.33
PAST-int-WT	77.46 / 6.59	31.99 / 34.30	11.77	11.29
PAST-int-WT($w = 0.25$)	79.13 / 6.96	32.54 / 35.43	12.06	11.60
PAST-int-WT($w = 0.75$)	67.76 / 5.98	27.33 / 30.35	10.69	11.22
PAST-int-FT($w = 0.25$)	74.35 / 6.40	30.47 / 32.13	12.65	11.61
PAST-int-FT($w = 0.5$)	70.66 / 6.13	28.66 / 29.76	13.06	11.89
PAST-int-FT($w = 0.75$)	65.57 / 5.85	26.05 / 28.45	12.15	11.98
PAST-sep-WT($w_p = 0.25$)	74.47 / 6.64	30.51 / 34.02	10.84	11.12
PAST-sep-WT($w_p = 0.5$)	72.14 / 6.27	29.04 / 32.07	10.90	11.10
PAST-sep-WT($w_p = 0.75$)	56.13 / 4.74	23.55 / 25.31	9.74	10.19
PAST-sep-CONCAT	74.22 / 6.48	30.33 / 33.26	11.37	11.03
PAST-sep-FT	40.67 / 3.35	18.08 / 19.01	6.92	9.31

表5 過去情報を含む文の数と情報の正しさ

モデル	文数	正解率 (%)
Gong ら	122	14.75
PAST-int-WT	99	14.14
PAST-int-WT($w = 0.25$)	99	10.10
PAST-int-WT($w = 0.75$)	94	14.89
PAST-int-FT($w = 0.5$)	94	15.96
PAST-int-FT($w = 0.25$)	93	15.05
PAST-int-FT($w = 0.75$)	95	10.53
PAST-sep-WT($w_p = 0.25$)	100	10.00
PAST-sep-WT($w_p = 0.5$)	78	14.10
PAST-sep-WT($w_p = 0.75$)	125	13.60
PAST-sep-CONCAT	139	16.55
PAST-sep-FT	92	20.65

表6 過去情報を含む文数の比較

モデル	文数	割合	総文数
リファレンス	11	3.4%	324
PAST-sep-FT	92	31.8%	289

果も示している。当日情報に関する自動評価ではGongらのモデルのスコアが高い一方、BLEUはPAST-int-FTのスコアが一番高い結果となった。表5に過去情報を追加したモデルにおける、過去情報に関する人手評価の結果を示す。Gongらのモデルは過去情報を入力していないため、過去情報への言及をランダムに出力していることになる。PAST-sep-CONCAT、PAST-sep-FTが他のモデルと比較すると正しい過去情報を出力している数と正解率が高い結果となった。

表6に、過去情報の出力の正解率が一番高かったPAST-sep-FTの出力を30試合抽出し、過去に言及している割合についてリファレンスと比較したときの結果を示す。この表を見ると、PAST-sep-FTはリファレンスに比べて過去に言及している割合が高く、リファレンスにはない過去情報を出力していることがわかる。

5 おわりに

本研究ではスポーツダイジェストの自動生成について、既存モデルの学習方法の変更による精度改善と、過去情報を含めたダイジェストが生成可能なモデルを提案した。既存モデルの損失関数の重みづけやファインチューニングをおこなうことで精度が向上すること、過去情報内容選択層を追加することで正しい過去情報に言及するようになることを示した。

参考文献

- [1] Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2253–2263, 2017.
- [2] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 6908–6915, 2019.
- [3] Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. Learning to select, track, and generate for data-to-text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2102–2113, 2019.
- [4] Li Gong, Josep Crego, and Jean Senellart. Enhanced transformer model for data-to-text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation (WNGT 2019)*, pp. 148–156, 2019.
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.
- [6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002.
- [7] Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 286–293, 2000.