DEIM Forum 2021 H11-5

新聞記事からのラジオ読み上げ原稿の自動生成

清水　嶺† 　酒井　哲也†

† 早稲田大学基幹理工学研究科情報理工・情報通信専攻 〒 1698555 東京都新宿区大久保 341
E-mail: †rei-shimizu@akane.waseda.jp, ††tetsuyasakai@acm.org

あらまし　通常，ニュースラジオ番組においては読み原稿が使用される．それは日経ラジオ社の場合でも例外ではなく，読み原稿は人の手によって作成されている．読み原稿の手動生成は，日経新聞の記事が元になっていることがほとんどであるため，特定のアルゴリズムを使用して自動生成することが可能であると考えられる．そこで本論文では，テキストセグメンテーションのアルゴリズムの一つである GraphSeg を利用した，教師なしの読み原稿自動生成アルゴリズムを提案する．実際に提案手法を実装したラジオ原稿自動生成ソフトを開発し，それを用いて生成したラジオ原稿を用いてユーザー実験を行った結果も示す．提案手法により自動生成した原稿と手動で作成したものとの side-by-side 評価を行った結果、両者には統計的有意差が認められないことが確認できた．

キーワード　自然言語処理，グラフデータ処理，テキスト分割

1 導　　入

日経ラジオでは，ラジオ NIKKEI と呼ばれるラジオニュースチャンネルを放送している [2]．通常，ニュースラジオの放送においては読み上げ用の原稿が用いられる．日経ラジオにおけるニュースコンテンツは日経新聞の記事 [1] に基づいて作成されている．本論文における試みは，日経ラジオの読み上げ原稿と日経新聞の記事間に存在する共通パターンを調べ，変換アルゴリズムを構築することで日経新聞の記事から自動的にラジオの読み原稿を生成することである．本アルゴリズムを実装した自動変換ソフトウェアを開発することで，読み原稿を作成するコストを大幅に削減することが可能となる．実際の新聞記事とラジオ原稿の異なる点としては，文章の長さ，敬体常体の違いが挙げられる. ラジオ NIKKEI の記事の場合，日経新聞の記事が元になっているケースが多く，記事のうち情報を伝えるにあたって重要なものを選出して構成されている．新聞記事の一般的な構成としては，前半部がニュースの概要，後半部がニュースをより深く理解するための周辺知識など，副次的な情報が与えられていることが多い. このため，ラジオ原稿自動生成の問題は，如何にして新聞記事の前半部を抽出するかという問題に帰着させることが出来る．つまり，記事を前半，後半に分けるアルゴリズムを構築する必要がある．前半がニュースの概要，後半がニュースの詳細であるという仮説に基づくならば，前半の最後のセンテンスと後半の最初のセンテンスでは大きく話題が転換していると言える．つまり，話題が共通している複数のセンテンスでまとまりを構築し，そのまとまり間の類似度に基づいてさらに大きなまとまりを構築し，結果的に 2 つのまとまりが出来上がるようにすればよい．本論文では，文のまとまりを構築するにあたって，Glavaš らによる GraphSeg [7] を基としたアルゴリズムを用いている．言い回しの変換は，林らによって提案されたルールベースの手法 [10] を用いた．また，ラジオ原稿とするにあたって不足している，新聞記事のメタデータから抽出された情報を元にした文の追加も行った．最後に，Side-By-Side な UI を持つ評価システムを開発し，実際に被験者に本物の読み上げ原稿と自動生成の原稿を比較して，どちらがより読み上げ原稿らしいかを判定してもらった．本実験では 12 名の被験者に実験に協力して頂き，得られたデータに対して検定を行った結果，両者には統計的有意差が見られないことが確認できた．

2 関 連 研 究

2.1 GraphSeg

線形テキストセグメンテーションは，一つのテキストから，意味の上で類似した文の塊の集合を構築する手法の一つである．これを用いて，文章の自動段落分けや自動要約を行うことが出来るため，自然言語処理や情報検索の分野では重要なトピックの一つである．教師なしテキストセグメンテーションのアルゴリズムとしては，Hearst による TextTiling [4] や Kehagias らによる動的計画法を用いた方法 [5] がある．GraphSeg と呼ばれるテキストセグメンテーションアルゴリズムは，教師なしであり，テキスト中のすべての文間の類似度を計算し，無向グラフを構築する．本アルゴリズムにおいては，類似度の計算手法として，単語ベクトルのコサイン類似度をベースとしている．S_1，S_2 を文，w_1，w_2 を単語とし，v_1，v_2 を対応する単語ベクトルとする．ただし，$A = \{(w_1, w_2) | w_1 \in S_1 \wedge w_2 \in S_2\}$ を満たす．このとき，類似度はこの類似度は以下の式で定義される．

$$sr(S_1, S_2) = \sum_{(w_1, w_2) \in A} \cos(v_1, v_2) \min(ic(w_1), ic(w_2)) \qquad (1)$$

なお，ここでの ic は自己情報量であり，以下の式で定義される．

$$ic(w) = -\log \frac{freq(w) + 1}{|C| + \sum_{w' \in C} freq(w')} \qquad (2)$$

ここでの $freq(w)$ は本文情報を含む大規模コーパス $|C|$ が与えられた時，w が生起する確率であるとする．しかし，上記の類似度指標は，文の長さに対してロバストではなく，文が長くなるとそれだけ値が大きくなるため，長さに関する正規化を行う必要がある．正規化を行った後の類似度指標は以下の式で表される．

$$rel(S_1, S_2) = \frac{1}{2}\left(\frac{sr(S_1, S_2)}{|S_1|} + \frac{sr(S_1, S_2)}{|S_2|}\right) \quad (3)$$

類似度がハイパーパラメータとして与えられる閾値を超えている場合は文がかなり類似している，としてノード間にエッジを貼る．このようにしてグラフを構築した後，グラフ中の最大クリークを求める．グラフ中の最大クリークというのは，任意の 2 ノード間にエッジが貼られているノードの集合のうち最大のものを指す．任意の 2 ノードにエッジが貼られているということは，最大クリークに含まれている文はすべてかなり類似している．つまり，何らかのトピックを持っていると判断することが出来る．最大クリーク問題は NP 困難であり，多項式時間で計算するアルゴリズムは見つかっていない．ここでは，最大クリークを求めるアルゴリズムとして Bron-Kerbosch アルゴリズム [6] を用いている．最大クリークの構築後，最大クリーク集合の中から，文章内で文が隣接しているものを抽出し，初期セグメントとする．初期セグメントの生成手順を図 1 に示す．図中の数字は，文章の中における文の位置を指している．例えば，1 であれば 1 番目の文である．

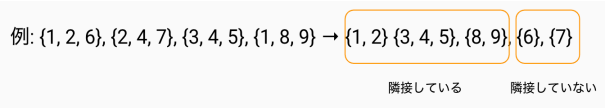

図 1　初期セグメントの生成

次に，構築された初期セグメントの併合を行う．ここでは隣接するセグメントの併合しか行わないが，あるセグメントに含まれている文が最大クリークに含まれており，その最大クリークに含まれている他のノードがあるセグメントに隣接するセグメントの中に含まれている時のみ併合を行う．この操作を経て構築されたセグメントを併合セグメントと呼ぶ．併合セグメントの生成を図 2 に示す．

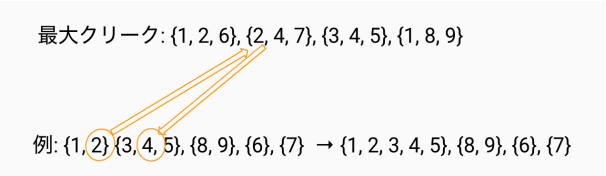

図 2　MergedSegment の生成

本アルゴリズムではハイパーパラメータとして，生成されるセグメントの最小サイズが与えられる．次に，含まれる任意のセグメントが最小サイズよりも大きくなるようなセグメントを構築する．ここでも，併合セグメントに含まれるセグメントのうち，隣接するものを併合することによって構築される．併合するセグメントを決めるにあたっては，セグメントの類似度を元にし，あるセグメントの前後のセグメントからより類似度の大きいセグメントと併合を行う．これによって構築されたセグメントを小セグメントと呼ぶ．SG_1，SG_2 をそれぞれ隣接するセグメントであるとした時，セグメントの類似度 sgr は以下の定義により成り立つ．式中の S_1，S_2 は，それぞれセグメント SG_1，SG_2 に含まれる文を指す．

$$sgr(SG_1, SG_2) = \frac{1}{|SG_1||SG_2|} \sum_{S_1 \in SG_1,\ S_2 \in SG_2} rel(S_1, S_2) \quad (4)$$

なお，ここでの rel は，3 を指す．

2.2　言い回しの変換

一般に，文字を媒体として情報を伝達する新聞記事と，音声を媒体として情報を伝達するのに利用されるラジオ読み上げ原稿は，語彙や言い回しが異なる．これは日経新聞の記事とラジオ NIKKEI の読み上げ原稿においても例外ではなく，読み上げ原稿の自動生成に際して，言い回しの変換を要求される．言い回しの変換はルールベースで行うことができ，その変換規則は林らによって提案されている [10].

3　提 案 手 法

3.1　ラジオ読み上げ原稿の生成手順

生成手順は以下のようになる．

（1） 文章を句点で分割．ここで分割された単位を文とする
（2） すべての文間の類似度を計算する
（3） グラフ閾値の計算，無向グラフの構築
（4） GraphSeg によるセグメントの生成
（5） セグメント数が 2 つになるように生成されたセグメントを併合する
（6） 言い回しの変換
（7） 欠損情報の補完

文間の類似度の計算においては，Word2Vec [8] モデルをベースとした類似度指標を用いるが，ここでは Word2Vec モデルとして，10 年分の日経新聞記事から学習したものを用いた．また，GraphSeg で使用する自己情報量を計算するために，KNB コーパス [9] を用いた．KNB コーパスは，構文・照応・評価情報つきのブログコーパスである．

3.2　GraphSeg の改良

GraphSeg は無向グラフを構築する際ノード間のエッジを張るか張らないかの閾値は，ハイパーパラメータとする．実際に適当に閾値を与え，新聞記事に対してテキストセグメンテーションを実行すると，ある新聞記事では最大クリークが求まらなかったり，適切なセグメントが得られない場合があった．グラフを構築する際の類似度を調べてみると，記事によって類似度の分布が異なっていることが分かった．つまり，ある記事ではほとんどのノード間のエッジが貼られるが，他の記事ではノー

ド間にほとんどエッジが貼られなくなるといった問題が発生する．そのため，記事によってグラフのエッジの閾値を自動で決定する仕組みが必要となる．提案手法では，記事のすべての文の類似度を計算した後に，類似度の中央値を取るように設定した．中央値を取ることで，任意の新聞記事に対して全結合グラフやノードが貼られないグラフが構築されることはなくなる．また，GraphSeg においては，最終的に構築されるセグメントの数を指定する機構が存在しない．つまり，最終的に構築されるセグメントの数が 2 つである必要がある本実験に適していない．そこで，提案手法においては，小セグメントが構築された後に，指定されたセグメントの数になるまで併合し続けるようにアルゴリズムを拡張した．ここで併合するか否かを決定する閾値は 4 を用いる．つまり，与えられたセグメントに対し，あるセグメントの前後に隣接しているセグメントとの類似度を計算した後，類似度が最も大きくなっているセグメント間を併合する．これを指定した数になるまで繰り返す．

3.3 欠損情報の補完

ラジオ読み上げ原稿においては，リスナーがより情報を正確に理解することを促進するため，特殊な文が挿入されている．例えば，ニュースが海外からのニュースであった場合，ラジオ読み上げ原稿では冒頭にニュースが得られた地点に関する情報が読み上げられることになる．例を表 1 に示す．この場合は，ニュースがニューヨークから得られた場合，「ニューヨークからのニュースです。」といった文が読み上げ原稿の冒頭に挿入される．一般に，新聞記事にはメタデータが付随している．特に，ニュースが得られた地点に関しては，記事の冒頭の【】で囲まれた位置に位置に関する情報が記載されている．提案手法においては，この【】で囲まれた情報から地点を抽出し，読み上げ原稿構築時に位置情報に関する文を挿入している．

表 1　ニューヨークから配信されたニュースの元になった記事の冒頭部とラジオ読み上げ原稿の冒頭の比較

ラジオ読み上げ原稿
ニューヨークからのニュースです。8 日のアメリカ株式市場でダウ工業株 30 種平均は 3 日続落し、 前の日に比べて 63 ドル 27 セント安い 2 万 5106 ドル 26 セントで終えました。 ....
元記事
【ＮＱＮニューヨーク＝川内資子】８日の米株式市場でダウ工業株 30 種平均は３日続落し、 前日比 63 ドル 27 セント安の２万 5106 ドル 26 セント（速報値）で終えた。 ....

記事の中には，記事に記載されている位置情報と，ラジオ読み上げ原稿に挿入される位置情報に関する表現が異なっており，直接文を作って挿入できない場合も存在する．例えば，上記の例の場合はそのまま位置情報を抽出すると，位置情報が「NQN ニューヨーク」となるが，実際に読み上げ原稿とするためには，「ニューヨーク」に変換されなければならない．そのため，提案手法では，事前に与えられた人手によるラジオ読み上げ原稿と，元になったニュースからその変換パターンを抽出し，パターンに沿った変換を行うようにした．

4 評　　価

4.1 実 験 設 定

本実験の評価は，ラジオ NIKKEI の読み原稿の作成に関わっていないユーザーを対象として行った．期間は 2 週間を設定し，12 人の日本経済新聞社の社員に協力して頂いた．実験の実行にあたって，50 件の自動生成の原稿を作成した．評価方法としては，人手によって作成したラジオ読み上げ原稿と，自動生成の読み上げ原稿を比較してもらい，どちらがより読み原稿として適しているかを判定してもらうというものであった．評価システムは Side-By-Side な UI を持ち，それぞれの記事が隣接するようになっている．

どちらが自動生成でどちらが人手なのかを悟られることを防ぐため，設問ごとにランダムに左右を入れ替えるような実装を行った．評価用アプリケーションはブラウザ上で動作するシングルページアプリケーションであり，サーバーサイドの API でデータを収集し分析するような実装とした．実際の評価システムの UI を以下の図 3 に示す．

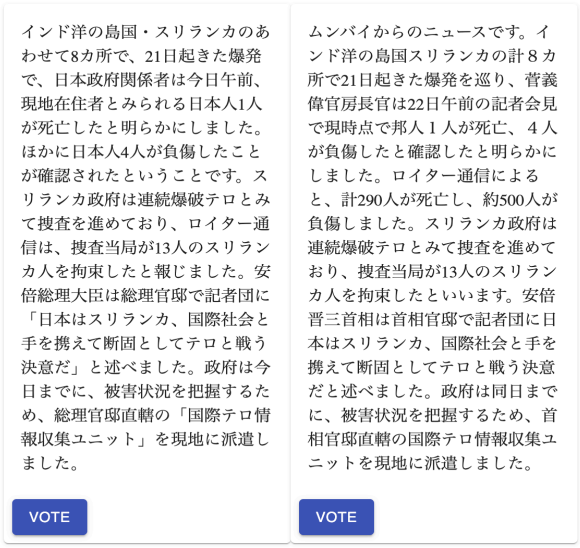

図 3　評価システムの UI

4.2 実 験 結 果

結果は以下のようになった．横軸を記事番号，縦軸を人数とした帯グラフを図 4 に示す．図 4 は，それぞれの記事データに対して，自動生成の読み上げ原稿と，人手によって作成された読み上げ原稿の比較を行った結果となっている．つまり，「どちらの記事がよりラジオ読み上げ原稿として優れているか」の判定を行った結果である，

提案手法により生成した原稿と人手により作成した原稿との差を調べるために，有意水準 5%におけるランダム化検定 [10] を行った．その帰無仮説は「自動生成した原稿と人手による原稿は同じ方法により得られたものである」となる．トライアル

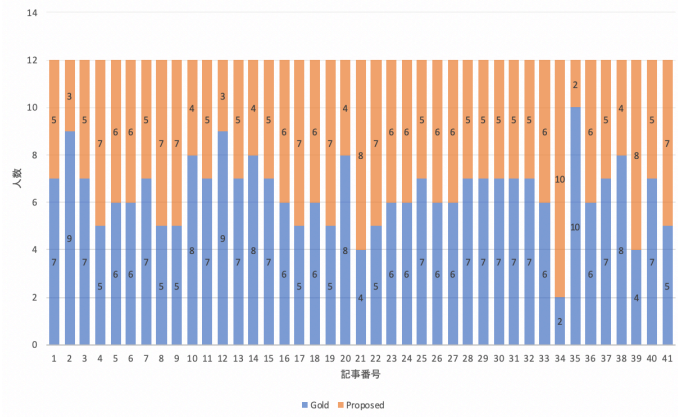

図 4　ユーザー実験の結果

数は 10000 で行った．ランダム化検定の図を図 5 に示す．行のセルの数は読み上げ原稿として適切であると判断した人数を表している．x_i は i 番目の記事について，提案手法が優れていると判断した人数であり，y_i は i 番目の記事について，人手によって作成されたものが優れていると判断した人数である．

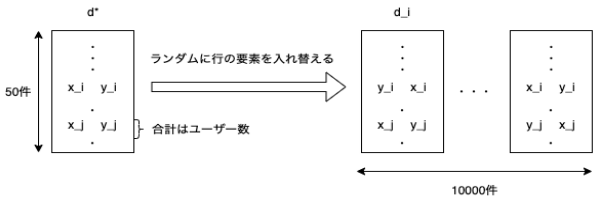

図 5　ランダム化検定

得られた p 値は 0.0906 であった．これにより帰無仮説が棄却できない事が分かり，勝ち負けについて統計的な有意差がないという事が分かった．つまり，提案手法により生成した原稿は，人手のものには及ばないものの、統計的には遜色ないという結果が得られた．

4.3　得られたデータの信頼性検証

評価者によって得られた評価結果が信頼できるものかどうかを検証するため， Krippendorff のアルファ係数 [12] を用いた．この指標は，2 人以上の評価者に対して適用することが出来る．Krippendorff のアルファ係数においては，比例尺度，間隔，順序，名義尺度のいずれかで計算することができる．本実験では名義尺度を用いた．実際の計算には R 言語の irr パッケージを用いた．結果，得られた値は –0.0254 となった．つまり，被験者による評価に関する意見は分かれていると言える．そのため，データの信頼性については保証出来ない．

5　考　　察

5.1　データの信頼性

Krippendorff のアルファ係数を用いてデータの信頼性を計算したところ，被験者の意見は分かれており，信頼性は低いことが分かった．各記事とその記事に対して「人手によって作成したもののほうが優れている」と判断した人の割合を分析した結果を図 6 に示す．

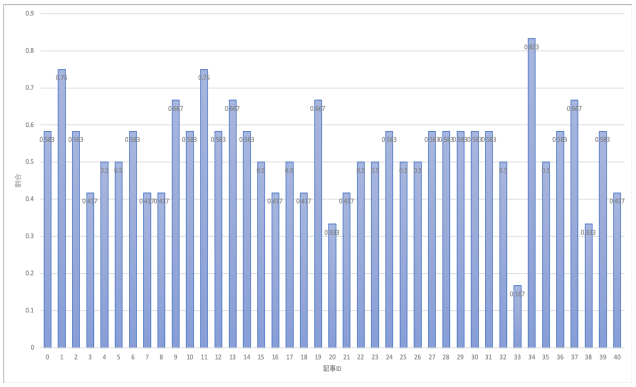

図 6　「人手によって作成したもののほうが優れている」と判断した人の割合

図 6 によると，多くの記事において，「人手によって作成したもののほうが優れている」と判断した人の割合が 5 割前後であることが分かる．つまり，多くの記事において，どちらが優れているかを明確に判定できなかったという事が分かる．原因としては，今回のユーザー実験では，多くのデータが取れなかった他，「どちらがよりラジオ読み上げ原稿として優れているか」といった曖昧な実験設定ししていたためこのような結果になったのではないかと考える．「優れている」というのは曖昧であり，優れていると明確に判断できる基準が被験者にはないため，このような結果になったのではないかと思われる．従って，信頼性の高いデータを得るには，より具体的な問題設定を行う必要があると考える．

5.2　生成された文章について

生成には成功しているものの，文章が崩れているため直接ラジオ読み上げ原稿として利用できないケースがある．その一例を表 2 に示す．「米連邦準備理事会の利下げに追随し、下押し圧力が強まる経済をます。」のように，文末が壊れており，明らかに言い回しの変換に失敗している箇所が存在する．形態素解析を行うと，「支える」という動詞が得られるが，この「支える」がこれ以上分解出来ないため，現在の実装ではうまく変換することが出来ない．つまり，「支える」から「支え」という単語を抽出し，「る」を「ます」に変換するような仕組みが必要であることが分かる．このようなパターンはいくつも存在するため，このような場合に対応できるようにシステムを改良する必要がある．

6　結　　論

本研究では，新聞記事からラジオ読み上げ用原稿を自動生成することを試みた．システムを開発し，ユーザーに実際の読み上げ原稿と自動生成の原稿の比較をしてもらった．その結果．自動生成のものは統計的には実際のものと比べて遜色ないことが分かった．ただ，考察においても述べたように，細かい部分に対応出来てないケースがあるため，完全自動化というわけには行かないのが現状である．今後の課題としては，完全自動化

表 2　自動生成に失敗した文章

北京からのニュースです。中国人民銀行は 20 日、事実上の政策金利である最優遇貸出金利の 1 年物をいまの 4.25 %から 4.2 %に下げると発表しました。 0.05 %の小幅な利下げとなります。米連邦準備理事会の利下げに追随し、下押し圧力が強まる経済をます。ＬＰＲは毎月 20 日に発表し、 最も信用度が高い企業に適用する貸出金利との位置づけ。形骸化していた以前のＬＰＲを衣替えし、８月から新たに公表を始めました。

を実現出来るようにするために，自動対応出来ないケースに対応する必要がある．システム自体は，このような変換規則を設定ファイルに記述して動作出来るような設計になっているため，変換に失敗している箇所を目視で発見し，手動で設定ファイルに追加し，ルールを蓄積すれば完全に近い自動化が実現出来るようになるのではないかと考える．

7 謝　　辞

論文を細部に渡って添削して頂いた河東宗佑先輩，提案手法に関して様々な意見をくださった諸先輩方，そして実験に協力していただいたり記事データを提供していただいた日本経済新聞社の皆様，並びにラジオの読み上げ原稿のデータを提供していただいた日経ラジオ社の皆様に多大なるお礼を申し上げます．

文　　献

[1] 日経電子版 トップページ，https://www.nikkei.com/
[2] ラジオ NIKKEI，http://www.radionikkei.jp/
[3] Alba,R．D., “A graph-theoretic definition of a sociometric clique“ The Journal of Mathematical Sociology．3，pp 113-126，1973
[4] Hearst，Marti A, “TextTiling: A quantitative approach to discourse segmentation“,
[5] Kehagias，Athanasios,Fragkou，Pavlina,Petridis，Vassilios, Linear Text Segmentation using a Dynamic Programming Algorithm, 10th Conference of the European Chapter of the Association for Computational Linguistics，apr 2003
[6] Bron,C．and Kerbosch,J., Algorithm 457: Finding All Cliques of an Undirected Graph [H], Communications of the ACM．16 pp 575-577，1973
[7] Glavaš，Goran and Nanni，Federico and Ponzetto，Simone Paolo，Unsupervised text segmentation using semantic relatedness graphs, Association for Computational Linguistics，2016.
[8] Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg S and Dean, Jeff, Distributed Representations of Words and Phrases and their Compositionality Advances in Neural Information Processing Systems 26, pp 3111–3119, 2013
[9] 橋本 力，黒橋 禎夫，河原 大輔，新里 圭司，永田 昌明, 構文・照応・評価情報つきブログコーパスの構築, 自然言語処理．18．pp 175-201，2011
[10] 林，由紀子，松原，茂樹，自然な読み上げ音声出力のための書き言葉から話し言葉へのテキスト変換, 情報処理学会研究報告．NL, 自然言語処理研究会報告．179 pp 49-54，may 2007.
[11] 酒井哲也，情報アクセス評価方法論　検索エンジンの進歩のために, コロナ社，pp 156-158，2015.
[12] Tetsuya Sakai, How to Run an Evaluation Task, Springer International Publishing, pp 17-20, 2019