芸術科学会論文誌 Vol. 18, No. 1, pp. 63 – 75 (2019)

ストーリー解析のための文の分散表現に基づく小説の自動セグメンテーション手法の提案

福田清人 [1)](学生会員)　森直樹 [1)](非会員)　松本啓之亮 [1)](非会員)　岡田真 [1)](非会員)

1) 大阪府立大学

A Novel Segmentation Method of Novels for Story Analysis based on the distributed representation of sentences

Kiyohito Fukuda[1)]　**Naoki Mori**[1)]　**Keinosuke Matsumoto**[1)]
Makoto Okada[1)]

1)Osaka Prefecture University

{fukuda @ ss., mori @, matsu @, okada @ }cs.osakafu-u.ac.jp

概要

近年, 計算機の爆発的な発展を背景として, 人工知能技術が大きく成長している. その中で小説や漫画, 絵本のような物語を自動生成する試みや, 人が物語を創作する際の創作支援を計算機にさせる試みが大きな注目を集めている. 自動生成や創作支援に関する研究は, 小説や漫画, 絵本などで数多くなされており, 有効な手法やシステムも報告されている. 一方, 創作物の理解に必要な物語の解析に関する研究は, 専門家の経験則に基づいて人手で情報を抽出するものが主流であり, 工学的な情報抽出や解析に関する研究はほとんど報告されていない. 以上を背景として, 本論文では文の意味を考慮した小説の自動セグメンテーション手法およびストーリーの解析手法を提案する. また, 実際の小説を用いた実験により, 提案手法を用いることで小説から意味のあるセグメントを自動抽出できることや作品間でのストーリー展開の類似性を計算できることを示した.

Abstract

Recently, automatic generation and creative support of narrative works like novel and picture book by the computer has attracted interest in artificial intelligence fields. Lots of studies on story generation and creative support have been reported. However, there are few studies on story engineering analysis by the computer. In this study, we propose the segmentation method of novels considering sentence meanings and story analysis method based on the distributed representation of sentences generated by bidirectional long short-term memory and autoencoder. To confirm the effectiveness of the proposed method, experiments are carried out by using some existing novels.

1 はじめに

近年, 計算機の爆発的な発展を背景として, 深層学習 (Deep Learning) に代表される人工知能技術が大きく成長してきた. その中で小説や漫画, 絵本のような物語を計算機によって自動生成する試みや, 人が物語を創作する際の創作支援を計算機にさせる試みが大きな注目を集めている. ここで, 物語とは人の感性に基づく創作物であり, ストーリーと表現媒体という 2 つの要素に分解することができる. ストーリーは物語の内容であり, 表現媒体は言語や画像のようなストーリーを表現するための媒体である. 本研究では, 表現媒体ではなくストーリーに焦点を当てる. これは, ストーリーの方が表現媒体よりも時間経過に対してロバストであり, 計算機の創作物理解にとって有用であると考えたためである. また, 今回は言語のみの単一表現で生成されている小説を対象とする.

自動生成や創作支援に関する研究は, 小説 [1] や漫画, 絵本 [2] などで数多くなされており, 有効な手法やシステムも報告されている. 一方, 物語の解析に関する研究は, 専門家の経験則に基づいて人手で情報を抽出するものが主流であり, 工学的な情報抽出や解析に関する研究はほとんど報告されていない. また, 文の意味のような深層特徴を用いた物語解析の研究もほとんどなされていない.

物語の自動生成や創作支援を実現するためには, 既存の物語を工学的に解析し, 人が物語を創作する上で必要な知識や技術を計算機が理解可能な形で獲得する必要がある. 具体的には, 機械学習技術に基づく既存の物語に対する工学的解析による有用な情報の抽出は必要不可欠な技術である. 人工知能の応用という観点からは, パターン識別や分布推定では人の創作物理解は困難なのではないかという批判もある. 人の感性に基づく創作物理解はもちろん容易ではないが, そこに踏み込んだ研究は人工知能研究の発展において最優先課題といっても過言ではない.

以上を背景として, 本論文では文の意味を考慮した小説の自動セグメンテーション手法およびストーリーの解析手法を提案する. 文の意味を考慮するために, Long Short-Term Memory (LSTM) および Autoencoder を用いた文の分散表現獲得手法を基礎として, 文の分散表現に基づき小説文をストーリーが展開する部分で自動分割する. また, 自動分割された複数の小説文からストーリー展開が類似した部分を発見する.

以下に本論文の構成を示す. 第 2 章では, 本論文と関連する研究について述べ, それらと本論文との位置づけについて述べる. 第 3 章では LSTM と Autoencoder に基づく文の分散表現獲得手法について説明し, 第 4 章では, 文の分散表現を用いた小説文のセグメンテーション手法およびストーリー展開に着目した小説文の解析手法について述べる. 第 5 章で実験により小説文に対するセグメンテーション手法および解析手法の有効性を確認する. 最後に第 6 章でまとめと今後の課題について述べる.

2 関連研究

本論文と関連のあるいくつかの研究について紹介し, それらの関連研究と本論文との違いを示す.

2.1 物語の自動生成に関する研究

自然言語を用いた物語の自動生成システムに MINSTREL[1] が存在する. MINSTREL は 事例ベース推論 (Case-Based Reasoning: CBR) に基づき, 登場人物の問題解決計画に作者レベルの問題解決計画を加えることで, アーサー王と円卓の騎士に材を取った小説を自動生成するシステムである. MINSTREL では, 既存の事例となる文章から CBR により文法的な面で置換可能な規則を推測する. 推測された規則を用いて文章を置換することで, 新たな物語を生成する. しかしながら, 置換可能であると推測された規則が意味的にも置換可能であるかは保証されておらず, 意味的に不自然な文章になる場合があることが問題点としてあげられる.

漫画の自動生成に関する研究では, 4 コマ漫画の絵の時系列的状態遷移に特定のパターンが存在するという仮定のもと, コマ間の状態変化に着目した絵モデルを用いて, ユーザの初期入力に基づいて 2 コマ漫画を自動生成するシステム [2] が提案されている. 絵モデルは絵がもつ意味情報や連続するコマ間の状態遷移を計算機で扱うためのモデルである. また, 描画オペレータと呼ばれる演算子によって状態遷移を定義している. しかしながら, 絵モデルでは状態間の因果関係が定義されていないため, 状態遷移の原因を明確に表現できない. さらに, 同一の絵中で発生している行動に対して, 順序が一意に決定できないことも問題点としてあげられる.

2.2 物語の創作支援に関する研究

小説の創作支援に関する研究では, 物語論に基づくストーリーテンプレートおよび物語の盛り上がりを表す想定感情線を用いてユーザが継続的に使用可能で物語の構造を整理することができるシステム [3] が提案されている. このシステムでは物語論で解析, 分類されている登場人物の役割やストーリー展開などの代表的なものをいくつか利用し, 創作に一定の方向性を与えることでユーザを支援する. また, 小説の各部分における感情値を可視化することで, ユーザが小説全体の流れを把握しやすくしている.

絵本の創作支援に関する研究では, ペタと呼ばれる絵本の半自動生成システムを用いてユーザが物語を創作するプロセスを考慮したストーリーの構成支援システム [4] が提案されている. ペタはイラストパーツを挿入して 1 ページ目の絵を作成し, その絵と関連のあるパーツをデータベースから抽出することで, 2 ページ目以降の絵を生成するシステムである. 提案システムではペタによって生成される絵に対して, 子供がストーリーを生成する際にシステムが「どんなおはなし？」や「なにしているところ？」といった 5W1H に沿った基本的な問いかけを実行する. 問いかけることで生成されるストーリーの構成を誘導し, ユーザを支援する.

しかしながら, どちらの研究についても物語を創作する主体はユーザであり, システムはユーザの補助をするのみである. システムが任意のストーリー展開をユーザに提示するような, 主体的な創作支援がないという問題点があげられる.

2.3 テキストセグメンテーションに関する研究

テキストデータをトピックなどの意味的なまとまりに分割するテキストセグメンテーションに関する代表的な手法に TextTiling[5] がある. TextTiling はテキスト中のある 2 文間を基準として, その前後の文をあらかじめ設定した窓幅の分だけそれぞれ取得し, 得られた前後の文章に対して単語の出現頻度ベクトルの類似度を計算する. この操作を基準となる 2 文を動かしながら実行し, 得られた類似度の変化から文章境界を推定する手法である. TextTiling は文章内に出現する単語に基づいてセグメンテーションするため, 短い文章を対象とした場合には有効に機能しないことが知られている. この問題点を解決するため, TopicTiling[6] と呼ばれる手法が提案されている. TopicTiling は単語の出現頻度ではなく, Latent Dirichlet Allocation (LDA) のようなトピックモデルで得られる各トピックに対する確率分布を利用して類似度を計算する手法である. トピックモデルを利用することで単語スパースネスの問題を解決できることが報告されている.

しかしながら, これらは単語ベースの手法であり, 文を基本単位としたテキストセグメンテーション手法の報告は少ない.

2.4 物語の解析に関する研究

物語の解析に関する研究では, 星新一の作品を構造分析の考えに基づきテキストの時系列に着目して物語のパターン抽出をする研究 [7] が報告されている. しかしながら, 物語のパターンを抽出するためにはテキストを抽象化して分類する必要があるため, 人手によってしか解析できないという問題点が存在する. また, 漫画に対してセリフ内容やフキダシの種類といった構成要素を用いて解析する研究 [8] も存在する. しかしながら, この研究では単語やオノマトペの使用頻度といった物語の表層的な特徴のみを用いており, 文の意味のような深層的な特徴を用いていないため, ストーリーのような深層的な情報を抽出することができないことが問題点としてあげられる.

2.5 本論文の位置づけ

上述した物語に関する研究では, 既存の物語やそのストーリーに関する人の知見やアノテートの結果を直接的な形で再利用することが多い. 本論文では既存の物語のストーリーを計算機のみで解析することでストーリーの関係性を理解し, ストーリーの関係性から新たなストーリーを生成する. また, ストーリーの展開を理解するために, 関連研究のような単語ベースではなく文を基本単位として, 文がもつ意味によってテキストをセグメンテーションする手法を提案する. 同時に, 文の意味を考慮した分散表現を用い物語のストーリーを解析することで, 物語がもつ深層的な情報を抽出する.

3 意味を考慮した文の分散表現の獲得

文に基づいて小説文をセグメンテーションしストーリー展開を解析するため, 文の意味を考慮した分散表現を獲得する必要がある. そこで本論文では, これまでに提案してきた LSTM に基づく Autoencoder を用いた

文の分散表現獲得手法 [9] を改良して文の分散表現を獲得する. 図 1 に文の分散表現の獲得手法の概要を示す.

3.1 Encoder と Decoder

文の分散表現の獲得手法には自然言語処理の分野で有効性が示されている Encoder-Decoder モデルを用いる. Encoder-Decoder モデルを, Encoder の入力と Decoder の出力を同一にした Autoencoder として用いる. モデル全体を Autoencoder にすることで, Encoder と Decoder を連結する中間表現がその文の特徴を抽出したものになると期待できる. 文がもつ多様で複雑な情報を抽出する必要があるため, Encoder には双方向の LSTM を用い, ネットワーク全体を多層構造にする. LSTM を多層構造にすることで各 LSTM 層の隠れ状態ベクトルに異なる情報が保存されることが期待される.

3.2 入力と出力のデータ形式

入力および出力には 1 文を形態素解析により分割した各単語の分散表現を用いる. 以下に入力および出力データの生成アルゴリズムを示す. なお, 以降単語の分散表現を単語ベクトル, 文の分散表現を文ベクトルと呼称する.

1. 入力となる 1 文 s を形態素解析することによって N 個の単語列 $w_1 w_2 \cdots w_N$ を取得する.
2. 文頭を表す記号 w_{S} および文末を表す記号 w_{E} をそれぞれ文頭および文末に付与することで単語列を $w_{\mathrm{S}} w_1 w_2 \cdots w_N w_{\mathrm{E}}$ とする. w_{S} および w_{E} を付与したため, 単語列全体のサイズは $N+2$ となる.
3. 単語列 $w_{\mathrm{S}} w_1 w_2 \cdots w_N w_{\mathrm{E}}$ の各単語 w に対して, 事前に学習した Word2Vec[10] により単語ベクトル v_w を獲得する.
4. Encoder における順方向 LSTM への入力データを $\mathcal{X}_{\mathrm{f}}$ とし, $\mathcal{X}_{\mathrm{f}}$ を文頭記号 w_{S} から文末 w_N までの単語ベクトルの集合として以下のように定義する.
$$\begin{aligned}\mathcal{X}_{\mathrm{f}} &= \{\boldsymbol{v}_{w_{\mathrm{S}}}, \boldsymbol{v}_{w_1}, \cdots, \boldsymbol{v}_{w_N}\} \\ &= \{\boldsymbol{v}_0, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_N\}\end{aligned}$$
5. Encoder における逆方向 LSTM への入力データおよび Decoder における LSTM への入力データを $\mathcal{X}_{\mathrm{b}}$ とし, $\mathcal{X}_{\mathrm{b}}$ を文末記号 w_{E} から文頭 w_1 までの単語ベクトルの集合として以下のように定義する.
$$\begin{aligned}\mathcal{X}_{\mathrm{b}} &= \{\boldsymbol{v}_{w_{\mathrm{E}}}, \boldsymbol{v}_{w_N}, \cdots, \boldsymbol{v}_{w_1}\} \\ &= \{\boldsymbol{v}_{N+1}, \boldsymbol{v}_N, \cdots, \boldsymbol{v}_1\}\end{aligned}$$
6. Decoder における LSTM の出力データを $\mathcal{Y}_{\mathrm{b}}$ とし, $\mathcal{Y}_{\mathrm{b}}$ を文末 w_N から文頭記号 w_{S} までの単語ベクトルの集合として以下のように定義する.
$$\begin{aligned}\mathcal{Y}_{\mathrm{b}} &= \{\boldsymbol{v}_{w_N}, \boldsymbol{v}_{w_{N-1}}, \cdots, \boldsymbol{v}_{w_{\mathrm{S}}}\} \\ &= \{\boldsymbol{v}_N, \boldsymbol{v}_{N-1}, \cdots, \boldsymbol{v}_0\}\end{aligned}$$

ここで, 上述したアルゴリズムにより生成された入出力のサイズはすべて同一となっており, $|\mathcal{X}_{\mathrm{f}}| = |\mathcal{X}_{\mathrm{b}}| = |\mathcal{Y}_{\mathrm{b}}| = N+1$ である.

3.3 モデルの学習と文ベクトルの獲得

文ベクトルの獲得手法の学習アルゴリズムを以下に示す.

1. 学習用の文集合から 1 文を抽出して s とする. $i=1$ とする.
2. s に対して 3.2 節で示した操作を実行することで学習用データ $\mathcal{X}_{\mathrm{f}}$, $\mathcal{X}_{\mathrm{b}}$, $\mathcal{Y}_{\mathrm{b}}$ を生成する.
3. 入力データ $\mathcal{X}_{\mathrm{f}}$ の i 番目の要素 $\boldsymbol{x}_i^{\mathrm{f}}$ を Encoder 部分の順方向 LSTM へ入力する.
4. 入力データ $\mathcal{X}_{\mathrm{b}}$ の i 番目の要素 $\boldsymbol{x}_i^{\mathrm{b}}$ を Encoder 部分の逆方向 LSTM へ入力する.
5. i を $i+1$ と更新する. その後, $i \leq N+1$ であれば 3 へ.
6. $\boldsymbol{x}_{N+1}^{\mathrm{f}}$ および $\boldsymbol{x}_{N+1}^{\mathrm{b}}$ を Encoder へ入力した際の双方向 LSTM の 2 つの隠れ状態ベクトルを得る. それらを結合して, 線形層を通すことで元の次元数に次元圧縮したベクトルを得る. このベクトルを初期状態ベクトルとして, Decoder 部分の LSTM に与える. $j=1$ とする.
7. 入力データ $\mathcal{X}_{\mathrm{b}}$ の j 番目の要素 $\boldsymbol{x}_j^{\mathrm{b}}$ を Decoder 部分の LSTM へ入力する. また, その際の Decoder の出力を $\boldsymbol{o}_j$ とする.
8. 出力データ $\mathcal{Y}_{\mathrm{b}}$ の j 番目の要素 $\boldsymbol{y}_j^{\mathrm{b}}$ と Decoder の出力 $\boldsymbol{o}_j$ との誤差を $L\left(\boldsymbol{o}_j, \boldsymbol{y}_j^{\mathrm{b}}\right)$ として求める.
9. j を $j+1$ と更新する. その後, $j \leq N+1$ であれば 7 へ.
10. 誤差逆伝搬法により, Decoder 部分で求めた最終的な誤差 $\sum_{i=1}^{N+1} L\left(\boldsymbol{o}_j, \boldsymbol{y}_j^{\mathrm{b}}\right)$ を最小化するようにモデルを学習する.

また, 以下に文ベクトルの獲得アルゴリズムを示す.

1. 文ベクトルを獲得したい文に対して 3.2 節で示した

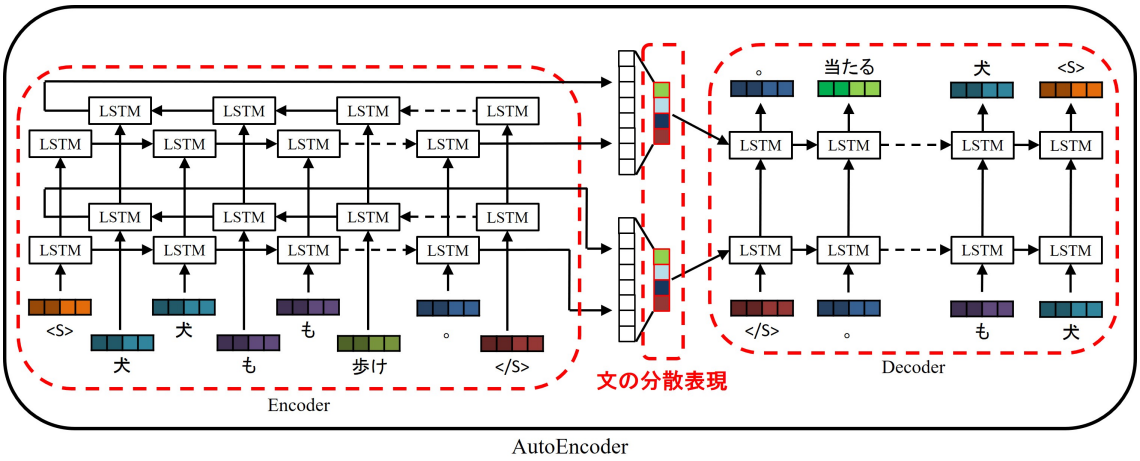


図 1　文の分散表現の獲得手法の概要

操作を実行することで, 入力データ $\mathcal{X}_{\mathrm{f}}$ および $\mathcal{X}_{\mathrm{b}}$ を生成する.

2. $i = 1$ とする.
3. 入力データ $\mathcal{X}_{\mathrm{f}}$ の i 番目の要素 $\boldsymbol{x}_i^{\mathrm{f}}$ を Encoder 部分の順方向 LSTM へ入力する.
4. 入力データ $\mathcal{X}_{\mathrm{b}}$ の i 番目の要素 $\boldsymbol{x}_i^{\mathrm{b}}$ を Encoder 部分の逆方向 LSTM へ入力する.
5. i を $i+1$ と更新する. その後, $i \leq N+1$ であれば 3 へ.
6. $\boldsymbol{x}_{N+1}^{\mathrm{f}}$ および $\boldsymbol{x}_{N+1}^{\mathrm{b}}$ を Encoder へ入力した際の双方向 LSTM の 2 つの隠れ状態ベクトルを得る. それらを結合して, 線形層を通すことで元の次元数に次元圧縮したベクトルを得る. このベクトルを文ベクトルとして獲得する.

4　提案手法

本論文では物語の中でもストーリーという要素に着目して小説を解析する. ここで, ストーリーをイベントや登場人物の行動, 場所移動に伴う物語中の一連の状態遷移の時系列だと定義する. 小説をいくつかの文章の集合であると仮定すると, ある連続した 2 つの文章間の変化が状態遷移であり, 冒頭から末尾までの連続した 2 文章の変化がストーリーであるといえる. そこで, 小説中の文章を分散表現化して文章ベクトルを得るとすると, 小説は文章ベクトルの集合とみなすことができる. 連続する 2 つの文章ベクトルに何らかの演算子を適用した結果がその 2 文章間での状態遷移を表しているといえる. そのため, 小説におけるストーリーは冒頭から末尾までの連続した 2 つの文章ベクトルにある演算子を適用した結果の集合であると言い換えることができる.

以上の観点から, 本章では 3 章で説明した文ベクトルを用いて, 小説文を対象としたテキストセグメンテーション手法およびストーリー展開の解析手法について説明する.

4.1　文の分散表現を用いた小説文のセグメンテーション

TextTiling では基準点に対して前後の窓幅内の単語集合についてベクトルを生成し, ベクトル同士の類似度が極小となる点を分割点と推測する. 本論文では, この TextTiling の考え方を基にして, 小説文の各文ベクトルに対して類似度を計算し, 類似度が極大となる 2 文を結合していく操作を, セグメント数が任意の数となるまで繰返すことでセグメンテーションする手法を提案する. ここで 1 文単位での類似度計算をすると, 機械的な文分割により分割されてしまった不適切な文の前後を分割点と推測してしまう可能性がある. そこである 1 文に対して, その 1 文と前後窓幅分を含む文ベクトルの平均を類似度計算に用いるベクトルとするスムージング手法を導入する. 図 2, 図 3 に小説文のセグメンテーション手法の概要およびスムージング手法の概要を示す. 以下に文ベクトルを用いた小説文のセグメンテーション手法のアルゴリズムを示す.

1. 獲得したいセグメント数を N_{s}, スムージングの窓幅を N_{w} とする. ここで, 本論文で用いるスムージン

グ手法では, 基準となる文に対してその前後の N_w 文を含む $2N_\mathrm{w}+1$ 文をまとめてスムージングする.

2. 解析する小説を M 文の文集合とする.
3. 小説中の各文に対して, 文ベクトルの獲得手法により文ベクトル $\boldsymbol{s}_i$ $(i=1,2,\cdots,M)$ を獲得する.
4. 各セグメントに対応したセグメントベクトルを $\boldsymbol{d}_j$ $(j = N_\mathrm{w}+1, N_\mathrm{w}+2, \cdots, M-N_\mathrm{w})$, セグメントベクトルの集合を $\mathcal{D} = \{\boldsymbol{d}_{N_\mathrm{w}+1}, \boldsymbol{d}_{N_\mathrm{w}+2}, \cdots, \boldsymbol{d}_{M-N_\mathrm{w}}\}$ とする. また, 各セグメントに含まれる文数を b_j, この文数の集合を $\mathcal{B} = \{b_{N_\mathrm{w}+1}, b_{N_\mathrm{w}+2}, \cdots, b_{M-N_\mathrm{w}}\}$ とする. ここで,

$$\boldsymbol{d}_j = \frac{1}{2N_\mathrm{w}+1} \sum_{k=j-N_\mathrm{w}}^{j+N_\mathrm{w}} \boldsymbol{s}_k \tag{1}$$

$$b_j = \begin{cases} N_\mathrm{w}+1 & (j = N_\mathrm{w}+1, M-N_\mathrm{w}) \\ 1 & (\text{otherwise}) \end{cases} \tag{2}$$

である.

5. セグメントベクトル集合の連続した 2 つのセグメントベクトル $\boldsymbol{d}$ および $\boldsymbol{d}'$ の類似度 $f_\mathrm{sim}(\boldsymbol{d}, \boldsymbol{d}')$ を以下の式に従って計算する. ここで, α は減衰率であり, セグメントが長文になりすぎないよう制御するための可調整パラメータである. また, b および b' はそれぞれセグメントベクトルに対応したセグメントに含まれる文数である.

$$f_\mathrm{sim}(\boldsymbol{d}, \boldsymbol{d}') = \alpha^{b+b'-2}\left(1 + \frac{\boldsymbol{d}\cdot\boldsymbol{d}'}{|\boldsymbol{d}||\boldsymbol{d}'|}\right) \tag{3}$$

6. 5 で求めた類似度が最大となった 2 つのセグメントベクトルを $\boldsymbol{d}_m$ および $\boldsymbol{d}_{m'}$ とし, それぞれに対応するセグメントに含まれる文数をそれぞれ $b_m, b_{m'}$ とする.
7. $\boldsymbol{d}_m$ および $\boldsymbol{d}_{m'}$ に対応するセグメントを結合し, 1 つのセグメントとする. その後, 以下の操作を適用することで各値を更新する. ここで, 記号 '$\to$' は右式を左式で更新する操作を表す.

$$\boldsymbol{d}_m = \frac{1}{2N_\mathrm{w}+b_m+b_{m'}} \sum_{k=m-N_\mathrm{w}}^{m'+b_{m'}+N_\mathrm{w}-1} \boldsymbol{s}_k \tag{4}$$

$$b_m \to b_m + b_{m'} \tag{5}$$

$$\mathcal{B} \to \mathcal{B}\backslash\{b_{m'}\} \tag{6}$$

$$\mathcal{D} \to \mathcal{D}\backslash\{\boldsymbol{d}_{m'}\} \tag{7}$$

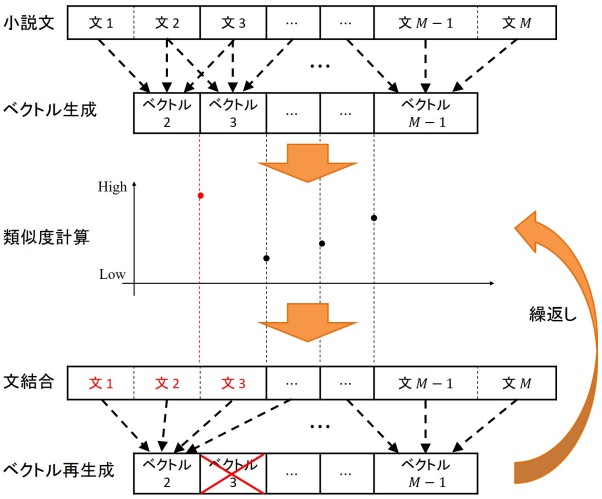


図 2　小説文のセグメンテーション手法の概要

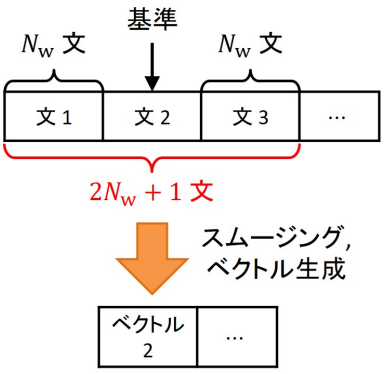


図 3　小説文のスムージング手法の概要

8. $|\mathcal{D}| > N_\mathrm{s}$ ならば, 5 に戻る.
9. $|\mathcal{D}| = N_\mathrm{s}$ の時, $\mathcal{D}$ および $\mathcal{D}$ の各要素に対応したセグメントを獲得する.

4.2　文の分散表現を用いたストーリー展開の解析

4.1 節で説明したセグメンテーション手法により得られたセグメントとそのセグメントに対応したセグメントベクトルを用いてストーリー展開を解析する. 以下にストーリー展開の解析手順を示す.

1. 解析対象とする作品を複数用意する.
2. 4.1 節のセグメンテーション手法により作品を自動分割し, 各作品のセグメントとそれに対応したセグメントベクトルを獲得する.
3. 各作品ごとに, 連続した 2 つのセグメントベクトルの差分を計算する.
4. 得られた差分ベクトルに対して, 他作品から得られ

た差分ベクトルとのコサイン類似度を計算する.

5. 得られたコサイン類似度や差分ベクトルの各要素の割合に基づいて, 作品間でのストーリーの類似性や, 差分ベクトルとストーリー展開の関係性などを可視化しつつ解析する.

5 実験

本章では提案手法の有効性を確認するため実験 1 ～ 3 をした.

5.1 実験 1

予備実験として, 文ベクトルの獲得手法により得られる文ベクトルが文のもつ意味を抽出できているかを確認する. 生成した文ベクトルを用いた類似度計算により, 意味的に類似した文同士の類似度が高くなれば, 有用な文ベクトルを生成できていると考えられる.

文ベクトルの獲得手法では, 入力データとして単語ベクトルを獲得する必要がある. 本研究では有効性が示されており, 実装が比較的容易な Word2Vec を単語ベクトルの獲得手法として利用する. 表 1 に用いた Word2Vec の設定を示す. 表 1 に記載していないパラメータについては今回実装に用いた, Keras2.1.5 のデフォルト値を用いた. Word2Vec 学習用のデータとして, 日本語 Wikipedia のテキストデータおよび, 小説投稿サイトである「小説家になろう」[11] から収集した各期間のランキング上位 100 件に含まれる 842 作品, 電子図書館である青空文庫 [12] から収集した, 著作権が切れた作家の 8689 作品を合わせた約 3.3GB のテキストデータを用いた. 頻出頻度の閾値以下の単語を未知語とした. 語彙数が指定した数より大きくなった場合には, 出現頻度の低い単語から未知語とした. Word2Vec の各パラメータは予備実験により決定した.

表 1 Word2Vec の設定

モデル	Skip-gram
高速化手法	Negative Sampling
文脈窓	10
ベクトルサイズ	200
サンプリングサイズ	15
バッチサイズ	6144
Epoch 数	50
最適化手法	Adam
初期学習率 α	0.0025
頻出頻度の閾値	10
語彙数	300000

5.2 実験 1 の実験手順

以下に実験 1 の手順を示す.

1. 小説投稿サイト「小説家になろう」 の “異世界 (恋愛)” および “現実世界 (恋愛)” という 2 ジャンルから小説を各 1 作品取得し, それぞれを 1 文に分割する. その中から会話文を除いた $n_{\min}$ 単語以上 $n_{\max}$ 単語以下の文をランダムに 100 文抽出する.
2. 青空文庫から夏目漱石の「吾輩は猫である」を取得し, 1 文に分割する. その中から $n_{\min}$ 単語以上 $n_{\max}$ 単語以下の文をランダムに 50 文抽出する.
3. 1 および 2 で抽出した文集合に対して, 文の一部分を人手で類義語や同意義の表現で置換することで, 類義文集合を生成する.
4. 「小説家になろう」 から類義文を生成する際に用いた作品およびその作品と同ジャンルの各 1 作品, 合計 4 作品を取得し, 1 文に分割する. その中から $n_{\min}$ 単語以上 $n_{\max}$ 単語以下の文をすべて抽出する.
5. 青空文庫から夏目漱石の「吾輩は猫である」および「こころ」の 2 作品を取得し, 1 文に分割する. その中から $n_{\min}$ 単語以上 $n_{\max}$ 単語以下の文をすべて抽出する.
6. 4 および 5 で抽出した文集合を候補文集合とする.
7. 文ベクトルの獲得手法を用いて, 候補文集合および類義文集合の文ベクトルを獲得する.
8. 候補文集合と類義文集合の類似度をコサイン類似度により計算する. 類義文を生成する際に元となった文が類似度の上位 n 文の中に含まれるかを確認し, 元となった文が含まれた場合, 正解とする.

5.3 実験 1 の実験条件

表 2 に文ベクトルの獲得手法の実験条件を示す. 実装には Keras2.1.5 を用い, 表 2 に記載していないパラメータについてはデフォルト値を用いた. 学習データには, 小説投稿サイト「小説家になろう」 から収集した各期間のランキング上位 100 件に含まれる 842 作品および青空文庫から収集した 8689 作品 の計 9531 作品に対して, それぞれを 1 文に分解した中の $m_{\min}$ 単語以上 $m_{\max}$ 単語以下の文のみを用いた. これは極端に短かっ

表 2　実験条件 1

$(n_{\min}, n_{\max})$	(12, 27)
$(m_{\min}, m_{\max})$	(10, 80)
Encoder 構造	2 層 双方向 LSTM
Encoder ユニット数	200
Decoder 構造	2 層 LSTM
Decoder ユニット数	200
バッチサイズ	1536
Epoch 数	5, 10, 15, 20, 25, 30
損失関数	平均 2 乗誤差
最適化手法	Adam
初期学習率	0.005

たり長かったりする文は学習データには不適だと考えたためである.

5.4　実験 1 の結果と考察

表 3 に実験 1 の結果を示す. ここで, 第 1 層とは入力データを受け取り第 2 層へと出力する層であり, 第 2 層は第 1 層の出力を入力として結果を出力する層である. また, epoch はモデルの学習回数であり, 学習に使用する全データを 1 度ずつ用いて学習すると 1 epoch となる.

表 3 を見ると, 上位 1 件のみに着目した場合, 5 epoch 分学習した時の第 2 層で得られた文ベクトルを用いて 127 という最大値を得た. また, 上位 5 件まで着目した場合は 10 epoch 分学習した時の第 2 層で得られた文ベクトルを用いて, 135 という実験における最大値を得た. 実験 1 における理論的な最大値は 150 であることを考えると, 十分に文の意味を考慮した文ベクトルが生成できているといえる. 今後の実験では 5 epoch 分学習した時の第 2 層で得られる文ベクトルを利用する.

5.5　実験 2

実験 1 から, 文の意味を考慮した文ベクトルを獲得できることがわかった. 実験 2 では, この文ベクトルを用いていくつかの小説を自動セグメンテーションし, 得られたセグメントについて考察することで提案手法の有効性を確認する.

表 4 に実験 2 の実験条件を示す. 実験 2 では青空文庫から取得した各作品を 1 文に分割し, その中から $m_{\max}$ 単語以下の文を用いた. 以降, 提案手法で得られたセグメントをシーンと呼称する.

表 3　実験 1 の結果

$n =$	1	2	3	4	5
第 1 層 (5 epoch)	104	109	110	110	110
第 2 層 (5 epoch)	**127**	**130**	**132**	**133**	134
第 1 層 (10 epoch)	99	105	107	108	109
第 2 層 (10 epoch)	124	**130**	**132**	**133**	**135**
第 1 層 (15 epoch)	100	102	104	105	106
第 2 層 (15epoch)	122	127	128	128	129
第 1 層 (20 epoch)	95	100	102	104	104
第 2 層 (20 epoch)	108	116	117	120	121
第 1 層 (25 epoch)	91	97	100	102	102
第 2 層 (25 epoch)	101	107	108	110	111
第 1 層 (30 epoch)	91	94	100	101	101
第 2 層 (30 epoch)	96	101	102	104	105

表 4　実験条件 2

$m_{\max}$	80
窓幅 N_{w}	1
分割シーン数 N_{s}	6
減衰率 α	0.985
使用作品	太宰治 「走れメロス」 太宰治 「黄金風景」 芥川龍之介 「蜘蛛の糸」 芥川龍之介 「藪の中」 エドガー・アラン・ポー 「黒猫」

5.6　実験 2 の結果と考察

表 5, 6 に実験 2 の結果, 生成されたシーンとその要約例およびシーンの本文全文の例を示す. ここで, 表 5, 6 内に書かれている “(unknown)” は, 文ベクトルの獲得手法に用いた Word2Vec 内で未知語として処理されてしまった単語である. また紙面の関係上, 走れメロスおよび蜘蛛の糸の各 1 シーンのみ全文掲載し, それ以外は中略した.

表 5 において, 「走れメロス」を分割して生成されたシーンについて内容を要約すると, 「村に戻ってきたメロスは妹と婿に無理を言って結婚式を挙げさせて幸せな時間を過ごす. その後メロスは未練を断ち切り王が待つ

町に戻ろうとする」である．また「蜘蛛の糸」を分割して生成されたシーンについては，「生前, 蜘蛛を一度だけ助けたことがあるカンダタを地獄から救い出してやろうと御釈迦様が蜘蛛の糸を地獄の底へ垂らす」と要約することができる．「黒猫」を分割して生成されたシーンについても，「殺してしまった妻を家の壁に隠した主人公は, 妻の死体と一緒に壁に隠してしまった猫のせいで犯行がばれてしまう」と要約できる．これらの結果から, 提案手法は小説からそれぞれのシーンを正確に分割することができるといえる．これは人手によるアノテートに頼ることなく実現されており, 人工知能による小説理解という観点からは画期的な結果である．しかしながら, 分割シーン数や減衰率の設定によっては 1 つのシーンとして分割されるべき部分が途中で分割されて 2 つのシーンとなってしまったり, 他のシーンと結合してしまうことが確認されている．作品ごとにこれらの可調整パラメータを最適化する方法について考察する必要がある．

5.7 実験 3

実験 2 から, 提案手法により小説内のシーンを分割できることがわかった．実験 3 ではシーン分割する際に得られたベクトルを利用して実際の小説に対して小説のストーリー展開を解析する．得られた解析結果に対して考察することで, ストーリーを工学的に解析するうえで今後必要となる知見を得る．

4.2 節で述べた方法で, 実験 2 に用いた小説に対してストーリー展開の類似性を解析する．実験条件は実験 2 と同様である．

5.8 実験 3 の結果と考察

図 4 および図 5 に実験 3 の結果, 類似度が比較的大きかったストーリー展開の差分を示す．横軸は差分ベクトルの要素のインデックスであり, 縦軸はその要素での差分の値である．また, 凡例は作品名とシーン番号を表す．例をあげると, 凡例が “merosu_1-2” であれば「走れメロス」の 2 番目のシーンと 1 番目のシーンの差分を表している．

図 4 は「走れメロス」の 2 番目のシーンと 1 番目のシーンの差分と「黄金風景」の 2 番目のシーンと 1 番目のシーンの差分であり, その類似度は 0.8182 である．そのストーリー展開について内容を確認すると, どちらも 1 番目のシーンは物語の冒頭であり, 物語内の時系列でも 1 番初めにあたる部分であった．2 番目のシーンについて,「走れメロス」 ではメロスが王様に激怒して城に乗り込んだ結果捕まってしまい, 親友であるセリヌンティウスを身代わりにして妹の結婚式のために村へと戻るシーンであった．「黄金風景」では主人公が子供時代にお慶という女中をいじめたという話から, 主人公が家を追い出され窮迫した後に病にかかってしまうというシーンであった．どちらのシーンにおいても主人公が感情を荒げる点や困った状況に追い詰められる点, 時間や場所などの環境が大きく変化する点など類似している点が多い．このことから, シーンの差分の類似度から, ストーリー展開の類似している部分を抽出することができるといえる．

図 5 は「藪の中」の 6 番目のシーンと 5 番目のシーンの差分と「蜘蛛の糸」の 4 番目のシーンと 3 番目のシーンの差分であり, その類似度は 0.6180 と比較的大きい．しかしながら, そのシーン展開について内容を確認すると,「藪の中」では, 5 番目のシーンと 6 番目のシーンでは同一の事件に対して, 異なる立場の人間が対立するような異なる供述をするという展開をしている．一方「蜘蛛の糸」では, 3 番目のシーンでカンダタは天から垂れる蜘蛛の糸を見つけ, これで地獄から抜け出せると喜んで登り始める．4 番目のシーンでは, 下を見たカンダタは他の罪人も蜘蛛の糸を登ろうとしていることに気が付き, 驚きと恐ろしさを感じる．このように同一人物の視点と感情が連動して変化するというストーリー展開になっている．この 2 つのストーリー展開はシーン間で意見や感情が対立的に変化するという点は同じである．しかしながら, 一方は同一の事象への立場の変化であるのに対して, 他方は対立する感情への同一人物の心情変化であり, 異なったものとなっている．このことから, ストーリー展開だけではなく文章構成の類似している部分も抽出することができるという知見が得られた．

図 4 および 図 5 を見ると, 差分が大きい要素はそれぞれ異なる．このことから, ストーリー展開や文章構成の違いによって, 変化が大きくなる要素は異なると考えられる．しかしながら, 本実験で解析した作品数は多くなく, より多数の作品に対して同様の実験をして, より詳細な解析をする必要がある．また, 単語ベクトルや文ベクトルを解析の基盤としているため, 得られた差分の各要素が 1 つの指標や意味に対応している訳ではない．そのため, 差分の各要素とストーリー展開や文章構成の種

表 5　生成されたシーンとシーン要約の例

作品名	シーン	本文	要約
走れメロス	1	メロスは激怒した。 必ず、かの (unknown) 暴虐の王を除かなければならぬと決意した。 (中略) 十里はなれた此の (unknown) の市にやって来た。	冒頭のシーン. 村の牧師であるメロスは今朝市にやって来て王に激怒した.
	2	メロスには父も、母も無い。女房も無い。十六の、内気な妹と二人暮しだ。 (中略) 呼吸もせぬくらいの深い眠りに落ちてしまった。眼が覚めたのは夜だった。	城に乗り込んだメロスは捕まり殺されそうになる. メロスは親友を身代わりに三日の猶予をもらい, 妹の結婚式を挙げるために村に戻った.
	3	メロスは起きてすぐ、花婿の家を訪れた。そうして、少し事情があるから、結婚式を明日にしてくれ、と頼んだ。 (中略) さらば、ふるさと。若いメロスは、つらかった。	メロスは妹と婿に無理を言って結婚式を挙げさせて幸せな時間を過ごす. その後メロスは未練を断ち切り王が待つ町に戻ろうとする.
黄金風景	1	海の岸辺に緑なす樫の木、その樫の木に黄金の細き鎖のむすばれて-(unknown)- (中略) 私は、(unknown) ことは嫌いで、それゆえ、(unknown) 女中を殊にもいじめた。	冒頭のシーン. 私は子供の時女中をいじめていた.
	2	お慶は、(unknown) 女中である。林檎の皮をむかせても、むきながら何を考えているのか、二度も三度も手を休めて、(中略) 「小説をお書きなさるんだったら、それはなかなか出世です」 私は苦笑した。	私はお慶という女中を特にいじめていたが, 一昨年家を追われ, 今では困窮して病にかかってしまった.
蜘蛛の糸	2	その命を無暗にとると云う事は、いくら何でも可哀そうだ。」と、こう急に思い返して、 (中略) 一すじ細く光りながら、するすると自分の上へ垂れて参るのではございませんか。	生前, 蜘蛛を一度だけ助けたことがあるカンダタを地獄から救い出してやろうと御釈迦様が蜘蛛の糸を地獄の底へ垂らす.
	3	(unknown) はこれを見ると、思わず手を拍って喜びました。この糸に縋りついて、どこまでものぼって行けば、 (中略) それからあのぼんやり光っている恐しい針の山も、足の下になってしまいました。	カンダタは蜘蛛の糸を登っていけば地獄から抜け出せると信じて登り始めた.
	4	この分でのぼって行けば、地獄からぬけ出すのも、存外わけがないかも知れません。 (中略) 折角ここへまでのぼって来たこの肝腎な自分までも、元の地獄へ逆落しに落ちてしまわなければなりません。	下を見たカンダタは他の罪人が糸を登ってきていることに気づき, 恐怖する.
藪の中	5	しかしそこに閃いていたのは、怒りでもなければ悲しみでもない、–ただわたしを蔑んだ、 (中略) 自分はいとしいと思えばこそ、大それた真似も働いたのだ、–盗人はとうとう大胆にも、そう云う話さえ持ち出した。	紺の水干の男が去った後, わたしは縛られた夫を殺し自分も死のうとしたが死ねなかった.
	6	盗人にこう云われると、妻はうっとりと顔を擡げた。おれはまだあの時ほど、美しい妻を見た事がない。 (中略) おれはそれぎり永久に、中有の闇へ沈んでしまった。……	妻が逃げ去った後, 盗人はおれを縛っていた縄を切ると立ち去った. おれは妻が落とした小刀を見つけると, それを胸に刺して自害した.
黒猫	6	壁には手を加えたような様子が少しも見えなかった。床の上の屑はごく注意して拾い上げた。 (中略) そいつのたてた声が私を絞刑吏に引渡したのだ。その怪物を私はその墓のなかへ塗りこめておいたのだった！	殺してしまった妻を家の壁に隠した主人公は, 妻の死体と一緒に壁に隠してしまった猫のせいで犯行がばれてしまう.

類との間にある関係性について解析することは非常に困難であり, 今後の課題である.

6　まとめと今後の課題

本論文では計算機による物語の自動生成を最終目的とし, その必須技術となる物語の工学的解析を実現するた

表 6　生成されたシーンの本文全文の例

作品名	シーン	本文
走れメロス	3	メロスは起きてすぐ、花婿の家を訪れた。そうして、少し事情があるから、結婚式を明日にしてくれ、と頼んだ。婿の牧人は驚き、それはいけない、こちらには未だ何の仕度も出来ていない、葡萄の季節まで待ってくれ、と答えた。メロスは、待つことは出来ぬ、どうか明日にしてくれ給え、と更に押してたのんだ。婿の牧人も頑強であった。なかなか承諾してくれない。夜明けまで議論をつづけて、やっと、どうにか婿をなだめ、すかして、説き伏せた。結婚式は、真昼に行われた。新郎新婦の、神々への宣誓が済んだころ、黒雲が空を覆い、ぽつりぽつり雨が降り出し、やがて車軸を流すような大雨となった。祝宴に列席していた村人たちは、何か不吉なものを感じたが、それでも、めいめい気持を引きたて、狭い家の中で、むんむん蒸し暑いのも忘え、陽気に歌をうたい、手を拍った。メロスも、満面に喜色を湛え、しばらくは、王とのあの約束をさえ忘れていた。祝宴は、夜に入っていよいよ乱れ華やかになり、人々は、外の豪雨を全く気にしなくなった。メロスは、一生このままここにいたい、と思った。この佳い人たちと生涯暮して行きたいと願ったが、いまは、自分のからだで、自分のものでは無い。ままならぬ事である。メロスは、わが身に鞭打ち、ついに出発を決意した。あすの日没までには、まだ十分の時が在る。ちょっと一眠りして、それからすぐに出発しよう、と考えた。その頃には、雨も小降りになっていよう。少しでも永くこの家に愚図愚図とどまっていたかった。メロスほどの男にも、やはり未練の情というものは在る。今宵呆然、歓喜に酔っているらしい花嫁に近寄り、花嫁は、夢見心地で首肯いた。メロスは、それから花婿の肩をたたいて、「仕度の無いのはお互さまさ。私の家にも、宝といっては、妹と羊だけだ。他には、何も無い。全部あげよう。もう一つ、メロスの弟になったことを誇ってくれ。」花婿は揉み手して、てれていた。メロスは笑って村人たちにも会釈して、宴席から立ち去り、羊小屋にもぐり込んで、死んだように深く眠った。眼が覚めたのは翌る日の薄明の頃である。メロスは跳ね起き、南無三、(unknown) たか、いや、まだまだ大丈夫、これからすぐに出発すれば、約束の刻限までには十分間に合う。きょうは是非とも、あの王に、人の信実の存するところを見せてやろう。そうして笑って磔の台に上ってやる。メロスは悠々と身仕度をはじめた。雨も、いくぶん小降りになっている様子である。身仕度は出来た。さて、メロスは、ぶるんと両腕を大きく振って、雨中、矢の如く走り出た。私は、今宵、殺される。殺される為に走るのだ。身代りの友を救う為に走るのだ。王の奸佞 (unknown) を打ち破る為に走るのだ。走らなければならぬ。そうして、私は殺される。若い時から名誉を守れ。さらば、ふるさと。若いメロスは、つらかった。
蜘蛛の糸	2	その命を無暗にとると云う事は、いくら何でも可哀そうだ。」と、こう急に思い返して、とうとうその蜘蛛を殺さずに助けてやったからでございます。御釈迦様は地獄の容子を御覧になりながら、この (unknown) には蜘蛛を助けた事があるのを御思い出しになりました。そうしてそれだけの善い事をした報には、出来るなら、この男を地獄から救い出してやろうと御考えになりました。幸い、側を見ますと、翡翠のような色をした蓮の葉の上に、極楽の蜘蛛が一匹、美しい銀色の糸をかけて居ります。御釈迦様はその蜘蛛の糸をそっと御手に御取りになって、玉のような白蓮の間から、遥か下にある地獄の底へ、まっすぐにそれを御下しなさいました。こちらは地獄の底の血の池で、ほかの罪人と一しょに、浮いたり沈んだりしていた (unknown) でございます。何しろどちらを見ても、まっ暗で、たまにそのくら暗からぼんやり浮き上っているものがあると思いますと、それは恐しい針の山の針が光るのでございますから、その心細さと云ったらございません。その上あたりは墓の中のようにしんと静まり返って、たまに聞えるものと云っては、ただ罪人がつく微な嘆息ばかりでございます。これはここへ落ちて来るほどの人間は、もうさまざまな地獄の責苦に疲れはてて、泣声を出す力さえなくなっているのでございましょう。ですからさすが大泥坊の (unknown) も、やはり血の池の血に咽びながら、まるで死にかかった蛙のように、ただもがいてばかり居りました。ところがある時の事でございます。何気なく (unknown) が頭を挙げて、血の池の空を眺めますと、そのひっそりとした暗の中を、遠い遠い天上から、銀色の蜘蛛の糸が、まるで人目にかかるのを恐れるように、一すじ細く光りながら、するすると自分の上へ垂れて参るのではございませんか。

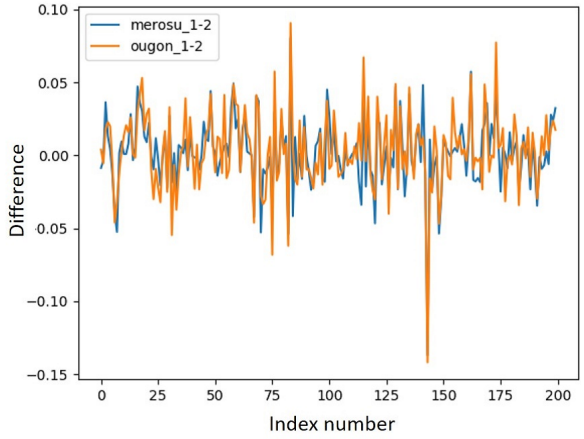


図 4　ストーリー展開の差分の例 1

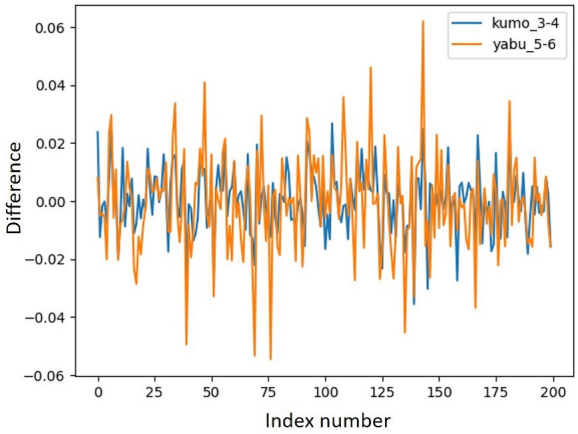


図 5　ストーリー展開の差分の例 2

め, 文の意味を考慮した文ベクトルに基づく小説の自動セグメンテーション手法とストーリーの解析手法を提案した. 自動セグメンテーション手法は TextTiling の考え方を基にした手法であり, ストーリーの解析手法はセグメンテーション手法で得られたセグメント間の差分からストーリー展開の類似性を解析する手法である. また, いくつかの実験により, 以下の知見が得られた.

- LSTM に基づく Autoencoder を用いた文ベクトルの獲得手法により得られた文ベクトルを用いることで文の意味を考慮することができる.
- 提案手法を用いることで, 人手によるアノテートに頼ることなく小説文を意味が考慮されたシーン単位に自動分割することができる.
- 提案手法により, ストーリー展開の類似性だけではなく, 文章構成の類似性も取得することができる.

今後の課題として, 階層的 LSTM や Attention 機構のような技術を導入することで文ベクトルの獲得手法の性能を向上させることがあげられる. また, 自動セグメンテーション手法における対象作品への可調整パラメータの最適化や, ストーリー展開の類似性に関するより詳細な解析手法の提案なども今後の重要な課題である.

なお，本研究は一部，日本学術振興会科学研究補助金基盤研究 (C) (課題番号 26330282) の補助を得て行われたものである.

参考文献

[1] Scott R. Turner. *Minstrel: A Computer Model of Creativity and Storytelling*. PhD thesis, University of California at Los Angeles, Los Angeles, CA, USA, 1993. UMI Order no. GAX93-19933.

[2] Miki Ueno, Naoki Mori, and Keinosuke Matsumoto. *2-Scene Comic Creating System Based on the Distribution of Picture State Transition*, pp. 459–467. Springer International Publishing, Cham, 2014.

[3] 葛井健文, 上野未貴, 井佐原均. 質問集合とグラフに基づく物語全体の流れを管理可能な創作支援システムの提案. 第 31 回人工知能学会全国大会発表論文集, 2016.

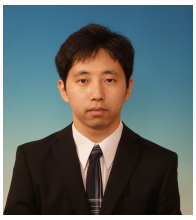
[4] 上原大輝, 出水ちあき, 宮里洸司, 神里志穂子, 野口健太郎. J-030 子どもの思考プロセス把握における物語自作システムの有効性検証 (hcs(2),j 分野: ヒューマンコミュニケーション&インタラクション). 情報科学技術フォーラム講演論文集, Vol. 10, No. 3, pp. 597–600, sep 2011.

[5] Marti A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, Vol. 23, No. 1, pp. 33–64, March 1997.

[6] Martin Riedl and Chris Biemann. Topictiling: A text segmentation algorithm based on lda. In *Proceedings of ACL 2012 Student Research Workshop*, ACL '12, pp. 37–42, Stroudsburg, PA, USA,

2012. Association for Computational Linguistics.
[7] 佐藤知恵, 村井源, 徃住彰文. 星新一ショートショート文学の物語パターン抽出. 情報知識学会誌, Vol. 20, No. 2, pp. 123–128, may 2010.
[8] 林沙輝, 中山伸一, 真栄城哲也. マンガの構成要素の定量的な解析と類似度判定. 第 75 回全国大会講演論文集, Vol. 2013, No. 1, pp. 845–846, mar 2013.
[9] Kiyohito Fukuda, Naoki Mori, and Keinosuke Matsumoto. *A Novel Sentence Vector Generation Method Based on Autoencoder and Bi-directional LSTM*, Vol. 800 of *Advances in Intelligent Systems and Computing*. Springer, 2019.
[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, Vol. abs/1301.3781, 2013.
[11] 小説家になろう - みんなのための小説投稿サイト. https://syosetu.com/.
[12] 青空文庫. https://www.aozora.gr.jp/.

福田 清人



2013 年大阪府立大学工学科卒業. 2015 年同大学大学院工学研究科電気・情報系専攻博士前期課程修了. 現在同大学院工学研究科電気・情報系専攻博士後期課程在学中. 主に自然言語処理や物語生成の研究に従事. 人工知能学会および芸術科学会の学生会員.

森 直樹

1968 年 9 月 2 日生. 1992 年京都大学理学部物理学科卒業. 1994 年同大学大学院工学研究科原子核工学専攻修士課程修了. 1997 年同大学院工学研究科電気工学専攻博士後期課程単位取得退学. 同年大阪府立大学工学部情報学科助手. 2005 年大阪府立大学工学研究科講師. 2007 年より大阪府立大学工学研究科准教授. 博士 (工学). 主に進化型計算, マルチエージェントシステム, 機械学習, 深層学習の研究に従事. システム制御情報学会, 電気学会, 計測自動制御学会, 日本シミュレーション&ゲーミング学会などの会員.

松本啓之亮

1978 年 3 月京都大学大学院工学研究科精密工学専攻修士課程修了. 同年 4 月三菱電機株式会社入社. 1996 年大阪府立大学工学部情報工学科教授. 現在, 大阪府立大学名誉教授. 主に知能情報処理やソフトウェアに関する研究に従事. 工学博士. 1983 年システム制御情報学会椹木記念賞論文賞, 1984 年電気学会学術振興賞論文賞, 2005 年電気学会学術振興賞進歩賞受賞. システム制御情報学会, 電気学会, 情報処理学会, IEEE などの会員.

岡田真

2001 年徳島大学大学院工学研究科知能情報工学専攻修了・博士 (工学). 2001 年大阪府立大学総合科学部数理・情報科学科助手, 同大学理学系研究科情報数理科学専攻助教を経て, 2012 年より同大学工学研究科電気情報工学専攻知能情報工学分野助教. 所属学会は電気学会, 人工知能学会, 言語処理学会, 情報処理学会. 専門分野は自然言語処理, 知識処理, 機械学習, 人工知能など.