小説内の動的人物相関図を用いた読書の支援

太田　彩 (東京工業大学），脇田　建（東京工業大学）

An Example of the Manuscript for  
the Visualization Symposium Japan (Times New Roman 16pt)

―Subtitle (Times New Roman 14pt)―

Taro KASHIKA and Hanako JOHO

##### ABSTRACT

This template is a guide to prepare manuscript for Visualization Symposium Japan. Here, please write the abstract serving as an index and as a summary of the present paper. It should be as long as approximately 150 words. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*《150語程度の英文ABSTRACT及び5語以下のKeywordsを記入してください．文字の大きさは10pt，字体はCentury，行間は１行分 (シングルスペース) です．》

Keywords: Visualization, Digital image processing, \*\*<5語以下>\*\*, \*\*\*\*\*, \*\*\*\*\*

# はじめに

# 関連研究(書きおわり)

## 小説からの情報抽出

小説の内容を理解する上で**登場人物の名前**と**人物間の関係**は大切な要素である。そのため以前から小説からそれらの情報を抽出するための研究がなされてきた。米田(米田崇明 et al. 2012)では人物候補の**局所出現性**と**述語情報**を利用し、小説の本文から登場人物を自動抽出した。西原(西原弘真 and 白井清昭 2015)は人物関係を表す語を集めた関係辞書と、人物間の関係を抽出する関係抽出のパターンの集合を作成し、人物間の関係を抽出した。しかし自然言語処理の技術だけを用いて正確な情報を抽出することは難しく、読者によって必要とする情報の範囲は異なる。そこで本研究では計算機による情報抽出に加えて、人間の手により情報の追加・修正を可能にすることにより、読書に有用な情報の提示を実現する。

## 物語の内容の可視化

物語の内容を一眼で理解したい、関係を図を用いて整理したいという願望は度々生じる。そのためテレビ番組や映画の内容を一枚の絵で表す研究は今までにも行われてきた。Filipovら(Filipov et al., n.d.)はテレビ番組「Game of Thrones」の中の84人の登場人物と216個の関係を放射状に配置することで一枚の絵で表した。Tanahashiら(Tanahashi, Hsueh, and Ma 2015)は時系列に沿って変わる映画内の登場人物の位置関係を複数の曲線を用いて表すアルゴリズムを提案した。以上の研究では物語の内容を一枚の絵で表しているが、本研究では読書の支援が目的であるため読み進めた時点までの関係を表示するインタラクティブな可視化を提案する。

## 文章に関する可視化

テキストに付随する情報をテキストに関連づけて表示する研究は行われてきた。Wuら(Wu, Zhao, and Amiruzzaman 2020)では音声の長さや音声認識の信頼度などの情報をテキストに関連づけて可視化をし音声認識のサポートを行った。Johnら(John et al. 2019)は文学者向けに登場人物の関係性の分析を可視化でサポートするシステムを提案した。本研究ではこれらの論文で採用されているいくつかの可視化手法を参考にしシステム設計を行う。

# 提案手法（書き途中）

読書をする際に読む意欲や必要性があっても、「登場人物が多すぎて覚えきれない」や「前回読んだ場所までの内容を忘れてしまった」などの理由から、読書を断念してしまうことがある。本研究ではこれらの読書の意欲を下げる要因を**読書の手間**と名付ける。読書の手間とは具体的には以下のような例が挙げられる。

* 前回読書をした時から時間が経ち内容を忘れてしまったので、また読み直す必要がある
* ずっと前に読んだ本の内容が思い出せない
* 登場人物が多い/名前が複雑で覚えられrない
* 登場人物間の関係が覚えきれない/どこで出てきた関係なのか忘れてしまった

これらの大半は記憶力を補助することで解決できると考え、本研究では小説の内容を可視化することで読書を支援することを決めた。

## 実験

人物相関図を用いて読書を支援する手法を考える上で、実際に手書きで人物相関図を書きながら読書を行った。その際に３つの問題を発見することができた。1点目は**一枚の絵で表すと重大なネタバレを生じてしまう**という問題である。人物相関図に物語内の登場人物や人間関係を表示すると、物語内の展開上重要な情報を事前に知ってしまう可能性がある。そのため読書をしたときの驚きや感動などの新鮮な感情が阻害され、読書の楽しみを減らす恐れがある。2点目に**人間関係の描写は複雑なものが多い**ということである。 小説には**親子**や**恋人**など単語で言い表せる関係もあれば、**〇〇さんの好きな人だと気づく**など簡潔に言い表せないものも存在する。文学において繊細な感情の変化は重要な要素なので詳細な関係を記述し、それを人物相関図にて確認できなければならない。3点目は**手書きで美しいレイアウトの人物相関図を描くことは難しいということ**である。登場人物や人間関係の数が多い場合、それらを線の交差や婉曲が少なくなるようにレイアウトすることには非常に手間がかかる。加えて読書を進めながら人物相関図を書くと、読みやすい人物相関図を書くことのハードルは跳ね上がる。そのため読書の度に人間の手でレイアウトを行うことは現実的でない。以上の考察により提案手法の果たすべき小目標を以下のように設定した。

-（P1) 小説のどの部分を読んでいても、人物相関図によってネタバレを起こさないこと -（P2) 小説内の繊細な人間関係を記載し、ユーザーがそれを確認できること -（P3) 入力した情報をもとに人物相関図を自動でレイアウトしてくれること

## 提案手法の概要

3.1で設定した小目標をもとに提案手法の概要を紹介する。 (P1)の実現のために本研究では**読書の進捗状況に応じて変化する人物相関図**を提案する。小説内の登場人物と人間関係を動的ネットワークとして扱うために、登場人物や関係の出てきた位置をページ数で管理する。そして今まで読んできた中に登場した人物・関係のみを人物相関図に表示することでネタバレのない可視化を実現する。(P2)の実現のために登場人物と人間関係の情報追加を手動と自動の２通りで実現した。自動抽出では独自のアルゴリズムにより登場人物を表す可能性の高い単語を抽出し、登場人物の共出現した文を関係として抽出した。手動での追加では登場人物と人間関係に関する様々な情報をユーザーの手によって登録する。(P3)の実現のために人物相関図の自動レイアウトを行った。レイアウトに必要な情報は現在までのページに出てきた登場人物と関係のリストである。それらの情報をもとに**人物相関図**と**人物相関表**という２種類の表示を行う。人物相関表はノードリンクを用いたグラフで、各ノードは登場人物を表し各リンクは人間関係を表している。人物相関表は隣接行列を用いたグラフで、登場人物は縦と横の軸に並んでいる。関係は有向であると考え、関係の元の人物と行かつ関係の先の人物の列であるセルを着色することで表現する。加えて本研究では**読書体験を共有するために、提案手法では登場人物や関係の情報を共有する**。そもそも登場人物のリストなどの読書に必要な情報は誰にとっても共通である。その情報を共有することで、新たに読む人は入力する手間を削減することができる。また、登場人物の間の関係は人によって受け取り方が異なることもあり、そこが議論の中心になることもある。 したがって関係を共有することで自身の細かい解釈についても人に伝えることができる。以上が提案手法の概要である。次の章では手法をもとにシステムの設計を行う。

## ユースケース

具体的なシステムの使用方法について考えるために、以下のようなユースケースを設定した。

大学生の花子さんは友人の太郎さんから勧められてイワン ツルゲーネフの「はつ恋」を読むことにした。彼女は読書をあまりしないため、中々読み進めることができず、システムを使って読書をすることを決めた。まず、青空文庫のサイトで「はつ恋」の図書カードを開く。そしてそこからリンクをコピーしシステムに貼り付けて、システム上で「はつ恋」を読めるようにした。

花子さんは読み進めると1ページ目で3人のおじいさんが出てくることに気づいた。それらの人物を登場人物に追加したいので、登場人物の候補をシステムに提示させた。その結果様々な候補が出されたので、ハイライトされた部分を確認しながら人物を表す単語のみを追加した。そのまま読み進めると5ページ目に「背の高い少女」が出てきたのでその人物を追加する。この時追加したい人物は一人だけだったので手動で追加した。加えてウラジーミル・ペトローヴィチは6ページ目で背の高い少女と知り合いになりたいと思っているので、その関係情報を追加する。その後 12ページ目で少女の名前が「ジナイーダ」であることが判明したので、先ほど入力した「背の高い少女」の名前を「ジナイーダ」に変更する。その後も読み進めたが、途中でロシア人の名前がたくさん出てきて読むのに疲れてしまったため、中断することにした。

1週間後花子さんは前回読み終わったところから読書を再開することにした。しかし期間が空いたために、登場人物の名前を忘れてしまった。そこでシステムにおいて人物相関図を表示して、内容を思い出すことにした。人物相関図を見ると今までに出てきた登場人物の一覧を見ることができ、特に重要な人物が「ウラジーミル・ペトローヴィチ」と「ジナイーダ」であることが確認できた。その後読み進めていくと「ルーシン」という人物が出てきた。「ルーシン」が誰か思い出せなかったので、システム上で「ルーシン」が本文中のどこに出ているのか確認し、思い出すことができた。

友人の太郎さんにそのシステムを使っていることを話すと、太郎さんは小説の考察をすることが好きなので、システムを使って考察をまとめてみることにした。先に花子さんが登場人物の情報を加えてくれていたので、太郎さんは手軽に関係を追加することができた。また各々自分が関係だと思った情報を追加していき、二人とも最後まで本を読み情報を入力することができた。その後二人でシステムを見ながら感想を語り合うことにした。システムを見ることで二人の解釈の違いや、印象に残ったシーンを気軽に見つけることができた。このように花子さんと太郎さんは読書体験を楽しんだ。

## システムの要件設定

ユースケースの内容からシステムの要件を以下のように定義した。

(N1) システムの上で読書ができること

(N2) 小説に関する重要な情報を手軽に追加できること

(N3) 情報を後から編集できること

(N4) 小説の進行度に合わせて関係図を表示すること

これらの要件を詳しく順番に確認していく。

### システムの上で読書ができること

この要件を設定した理由は主に２点存在する。1点目は読書における可視化の効果を高めるためである。読書をするスペースと可視化の画面が別れている場合、小説の本文と可視化を同時に確認することは難しく、可視化を見ることが負担になってしまう可能性がある。そのため読書スペースと可視化を同じシステム上で表示することでユーザーに積極的に可視化を活用してもらう。２点目は(N2)で定義した、情報の追加にかかる負担を削減するためである。読書の進度に合わせた可視化を行う上で「現在どこのページを読んでいるのか」や「登場人物や関係性がどこのページで出てきたか」などのページ数に関する情報は非常に重要である。しかしユーザーが読書を進めるたびにそれらの情報を提示してもらうことはあまり現実的ではない。そこでシステム上で読書をしてもらうことで自動的に現在のページ数を管理し、情報を追加してもらう際にも自動でページ数の情報を補足する。これらによってユーザーのシステム利用にかかる負担を削減する。以上がこの要件を設定した理由である。

### 小説に関する重要な情報を手軽に追加できること

２章において、小説における登場人物や関係性の認識は非常に曖昧であり機械による自動抽出が難しいことを議論した。そのため本研究では機械で抽出した情報を人間の手で修正し、小説の理解に適した登場人物と関係のリストを作成する手法を提案した。その手法を実現するためにシステムには**小説の本文から情報を自動抽出する**機能と**人間による情報追加を可能にする**機能の２つの機能が必要である。この機能を実装する上で最も重視すべき事項はこの機能がユーザーによって使いやすい機能であるかという点である。本研究の目的は**読書を円滑にする**ことであるため、情報の追加に手間取り読書を停滞させることは本意ではない。そのためなるべく少ない手順で情報を追加できるように機能の詳細を決定していく。

### 情報を後から編集できること

この機能は**謝った情報を入力した**とき、**機械による自動抽出にミスが存在した**とき、**読書が進むうちに情報が変化した**時などにおこる問題を解決するために必要である。システムを手動で操作したり、アルゴリズムで情報を自動抽出する時にミスが起こらないと考えることはできない。しかし誤った情報をそのまま保存しておくと映し出された人物相関図も誤った情報を含み、人物相関図を見ることで読解を阻害する可能性がある。そのため誤りを見つけた際には即座に修正できる必要がある。加えて小説は途中で情報が補完される場合がある。例えばユースケースにおいて序盤に出てきた「背の高い少女」は後に「ジナイーダ」という名前であることが判明した。名前が明かされた後、人物相関図における人名を表示を「背の高い少女」ではなく「ジナイーダ」に変えたいと思う場面も出てくるはずだ。そのためこの場合にもすでに入力した情報を修正できる必要がある。以上の理由から**情報を後から修正できること**をシステムの要件として設定した。その要件を達成するためにも、情報を追加するときと同様にユーザーが情報を修正しやすいような画面の配置、機能の設計が重要である。

### 小説の進行度に合わせて関係図を表示すること

この要件は（P1)と(P3)を達成するために非常に重要であり、この４つの中で最も大切な要件である。

以上の要件を元にシステムに必要な機能を実装した。４章においては実際のシステムについて詳しく説明を行う。

# 可視化システム（一旦書きおわり）

4.1にて設定した要件をもとにシステムを設計した。この節では要件に沿ってシステムの説明を行う。

## (N1)システムの上で読書ができること

様々な電子書籍リーダーを参考にし、読書に必要な機能を最低限実装した。読書画面には、小説の選択機能、本文の表示、しおりなどの機能が実装されている。**小説を選択**ボタンを押すと新しく小説を青空文庫のサイトからダウンロードするか、既にダウンロードした小説から選ぶのか選択することができる。小説をダウンロードする時は、青空文庫の作品の図書カードページのリンクを貼り付ける。貼り付けると、青空文庫のデータを表示しやすい形へ成形しシステム上に保存する。ダウンロード済みの作品を選ぶ際は、作品の一覧から自分の読みたい作品を選択する。選択された小説の本文はシステムの左下に表示され、 最初に読むときは１ページ目が、2回目以降に読む場合は前回最後に読んだページが表示される。元々の青空文庫で公開しているデータには区切りがないが、そのままシステム上に表示すると読んだ部分の把握が難しくなるので、一定文字数で区切りページとして管理している。 システムの上部には読書を進めるためのツールが位置している。左端のスライダーではつまみを動かして、見たいページを選択しそのページに飛ぶことができる。「◁」を押すと前のページへ、「▷」を押すと次のページへとめくることができる。次の「しおり」と書かれたボタンを押すと現在表示しているページをしおりづけできる。最後に「しおりに飛ぶ」ボタンを押すと、以前に追加したしおりのページを選択しそのページに飛ぶことができる。

## (N2)情報の追加方法

この節では読んだ小説内の情報をシステムに追加する方法を紹介する。このシステムで入力できる情報はで議論した通り、登場人物と関係性の二つである。システム上で入力された情報はシステムを閉じても保存され続ける。情報追加は手動と自動の二通りで行えるので順番に紹介する。

### 自動抽出

自動抽出は**自動情報抽出**ボタンを押すことで、小説内の登場人物と人間関係の候補を自動で全て抽出することができる。抽出された情報は自動で保存され、その情報はシステム内で確認することができる。

### 手動抽出

登場人物の追加方法は主に二つである。一つ目の方法は完全に手作業で追加する方法である。1 番上の**情報追加**ボタンを押すと、人物を追加するか、関係を追加するか選ぶことができる。人物の追加を選ぶと、情報入力画面が出る。テキストボックスに追加したい登場人物を入力し、**以上の人物を追加**ボタンを押すと、入力した人物が登場人物のリストに追加される。二つ目の方法は計算機を用いて登場人物の候補を提案し、その中から人物を選んでもらう方法である。 **登場人物の候補を抽出**ボタンを押すと、現在表示している文章内の登場人物らしい単語を抽出して表示する。候補のうちチェックを付けたものは文中での色が変化するので、その単語が本当に登場人物を指すのか確認できる。最後に登場人物として追加したいものだけチェックをつけて、**追加**ボタンを押すとまとめて登場人物に追加される。

次に関係性の追加方法について紹介する。システムの上部の**情報追加**ボタンを押し、**関係性追加**を選択すると関係性の入力フォームが出現する。そこでその関係が**誰から**、**誰への**、**どんな関係**で**どのくらい好意的なのか**を入力する。好感度は５段階評価で高い値ほど好感度が高いことを示す。以上の情報を入力し終わったら**追加**ボタンを押すことで入力した関係性はシステムに保存される。

## (N3) 情報を後から編集できること

この節では既に追加した情報を修正する方法について紹介する。情報編集は**情報編集画面**を開くことで行うことができる。情報編集画面は**登場人物編集表**と**登場人物の出現度のヒートマップ**と**関係編集画面**の3つから成り立つ。登場人物の情報を編集する場合は表において変更したい部分をダブルクリックし直接書き換えることができる。しかし書き換えただけではシステムに保存された情報は変わらないので右にある**変更を保存**ボタンを押す必要がある。加えて登場人物の左横にチェックをつけると登場人物が選択され、選択された人物はヒートマップ上に表示される。右の**消去**ボタンを押すとチェック付きの人物をまとめて消去することができる。これは関係編集画面も同様に操作することができる。 ヒートマップにおいて縦軸は登場人物、横軸はページ数を表している。各セルはそのページにおいて登場人物の単語が何回出てきたかによって色分けされており、濃い色であるほど登場回数が多い。カーソルを各セル上におくと、そのセルが「誰のどのページでの出現回数」を表していて、正確な出現回数を確認することができる。加えて各セルをクリックするとそのセルの表すページへ飛ぶことができる。

## (N4) 小説の進行度に合わせて関係図を表示すること

最後に関係図を表示する機能について紹介する。本研究では登場人物と関係を表すために**人物相関図**と**人物相関表**という２種類の表示を行う。

### 人物相関図

人物相関図はノードリンクである。人物相関図は上の**可視化を選択**から**人物相関図**を選択することで表示できる 。人物相関図には現在のページまでに出てきた登場人物と関係の情報が表示される。各ノードは登場人物を表し、各リンクは1番直近の関係性を表している。各ノードの大きさはその人物の持つ関係性の多さを表し、大きいほど小説において重要な人物である。ノード上でクリックをするとそのノードに繋がったリンクのみが強調され、関係する人物を一眼で確認できる。加えて各リンクはその関係の持つ好感度の値によって色付けされている。各リンクの上をホバーすると関係についての正確な情報を得ることができる。人物相関図は登場人物が少ない場合に一目で重要な人物や関係性を確認することができ、多くのユーザーに馴染み深い表示手法を採用しているため内容を理解しやすいという利点がある。

### 人物相関表

次に人物相関表の紹介を行う。人物相関表はシステムの上部の**可視化を選択**から**人物相関表**を選ぶことで表示できる。人物相関表では現在のページまでに出てきた人物の名前が縦軸と横軸に並ぶ。各セルは縦軸の人物から横軸の人物への直近の関係を表しており、その関係の好感度によって色分けされている。またセル上をホバーすることでその関係の詳細を確認でき、クリックすることで過去の関係の一覧表を見ることができる。人物相関表の隣には表示する範囲を制限するためのボタンが位置する。これらのボタンでは、**小説のどのページの関係を表示するのか**、**どの人物に注目した人物相関表を表示するのか**、**人物相関表にどの人物を表示するのか**の３つを調整することができる。 人物相関表の機能は以上の通りである。人物相関表の長所は登場人物の多い長い小説に対しても、機能を使えばある程度対応することができることである。 このように人物相関図と人物相関表では長所が異なっているため、人物の数や知りたい情報によってビューを使い分けることを想定している。

# 議論

この章では３章で定義した提案手法の果たすべき小目標を達成できているのか議論を行う。小目標は以下の通りであった。 -（P1) 小説のどの部分を読んでいても、人物相関図によってネタバレを起こさないこと -（P2) 小説内の繊細な人間関係を記載し、ユーザーがそれを確認できること -（P3) 入力した情報をもとに人物相関図を自動でレイアウトしてくれること

## （P1) 小説のどの部分を読んでいても、人物相関図によってネタバレを起こさないこと

本研究ではこの小目標を達成するために**読書の進度によって変化する動的人物相関図**を提案した。提案手法では登場人物や人物間の関係性は小説のページが進むにつれて登場しその後持続して存在するものとして扱う。そして現在表示しているページまでに出てきた登場人物と関係性を元に**人物相関図**と**人物相関表**という２種類の可視化を行う。これによりユーザーは表示するページ数の範囲を指定して人物相関図を表示することができ、ネタバレを受けずに書籍の内容を整理することができる。一方課題点は2点ある。1点目はシステムの誤操作によって意図せずまだ読んでいない部分の情報を取得してしまう可能性があることである。今回人物相関図表示する内容は「現在表示しているページまでに出てきた情報」を表示しているため、スクロールバーの操作を間違えてまだ読んでいない部分を選択するとその時点までの登場人物と関係を人物相関図に表示してしまう。今後の課題としては意図しない表示が行われないための仕組み作りがあげられる。2点目は人物相関図に表示された登場人物からネタバレを起こす可能性がある点である。現在一人の登場人物に対し名前が一つ対応しており、それは何度でも修正することができる。加えて小説内の関係情報は他者と共有されているため、先に読んだ人が自分の読んだ内容に合わせて登場人物の名前を修正している場合、登場人物の真の名前が人物相関図に表示される可能性がある。登場人物の名前がわざと序盤において隠されていたり展開に大きく影響している場合があるため、登場人物の真の名前を知ることでネタバレにつながる可能性がある。そのため今後の課題として

## （P2) 小説内の繊細な人間関係を記載し、ユーザーがそれを確認できること

この小目標を達成するためにシステムの要件として\*\*\*\*、\*\*\*\*

## （P3) 入力した情報をもとに人物相関図を自動でレイアウトしてくれること

Filipov, Velitchko, Davide Ceneda, TU CVAST, Michael Koller, Alessio Arleo, and Silvia Miksch. n.d. “The Circle of Thrones: Conveying the Story of Game of Thrones Using Radial Infographics.”

John, Markus, Martin Baumann, David Schuetz, Steffen Koch, and Thomas Ertl. 2019. “A Visual Approach for the Comparative Analysis of Character Networks in Narrative Texts.” In *2019 IEEE Pacific Visualization Symposium (PacificVis)*, 247–56. <https://doi.org/10.1109/PacificVis.2019.00037>.

Tanahashi, Yuzuru, Chien-Hsin Hsueh, and Kwan-Liu Ma. 2015. “An Efficient Framework for Generating Storyline Visualizations from Streaming Data.” *IEEE Transactions on Visualization and Computer Graphics* 21 (6): 730–42.

Wu, Tsung Heng, Ye Zhao, and Md Amiruzzaman. 2020. “Interactive Visualization of AI-Based Speech Recognition Texts.” *EuroVA*.

米田崇明, 篠崎隆宏, 堀内靖雄, and 黒岩眞吾. 2012. “述語情報を利用した小説の登場人物の抽出.” *言語処理学会第 18 回年次大会発表論文集* 18: 855–58.

西原弘真, and 白井清昭. 2015. “物語テキストを対象とした登場人物の関係抽出.” *言語処理学会第 21 回年次大会発表論文集* 21: 628–31.