物語の内容想起支援インタフェースの開発

田中 翔太郎[†1,a)] 岡部 誠[†1,†2,b)] 尾内 理紀夫[†1,c)]

概要：本稿では，物語の内容を効率良く想起するインタフェースの開発について報告する．単語の出現頻度や単語間の関係を視覚化することで内容理解を支援するシステムは多数存在するが，新しい人物が現れたり，人物関係が変化したりするといった物語の特徴を考慮したシステムは存在しない．ユーザは，テキストデータを入力し，自動で抽出された人物情報をマウスで編集することで，登場人物の関係図や人物の入った文を閲覧できる．本インタフェースを使用することで，過去に読んだ本を最初から読み返すことなく内容を想起できる．

キーワード：物語，関係図，ユーザインタフェース，人物抽出

An Interface for Reminding of the Content of the Story

Tanaka Shotaro[†1,a)] Okabe Makoto[†1,†2,b)] Onai Rikio[†1,c)]

Abstract: In this paper, we developed an interface to support reminding of the content of the story efficiently. There are many systems that can help us understand the content by visualizing the word relations and word appearance frequency. However, there is no such system that focused on the characteristics of the story, e.g. new character appearance, changes on relationships. First, the user inputs text data, then edits characters' names which are extracted automatically by the system. As a result, the user can browse the relationship diagram or sentences including the character's name. By using our developed user interface, we can remind of the content of the story easily than just simply reading them again.

Keywords: Story, Relationship Diagram, User Interface, Person Name Extraction

1. はじめに

過去に読んだ物語の内容を想起したいとき，最初から読み返すのは時間がかかる．実際，物語は短編でも 3 万字程度あり，長編では 35 万字にもなる．読了時間の目安としては，小説投稿サイト [*1] が毎分 500 字読めると想定して計算している．また，読了時間推定システムの研究 [5] は，1 万字に対する読了時間がユーザや読む作品ごとに異なるものの概ね 10 分から 20 分であると示した．これは，毎分 500 字から 1000 字に相当する．つまり，35 万字の長編を読むためには一般に 5 時間 50 分から 11 時間 40 分を要する．

そこで我々は，物語の内容を効率良く想起するためのインタフェースを開発した．ユーザは，自動で抽出された人物情報をマウス操作で編集することで，登場人物の関係図を閲覧できる．登場人物の関係図は，ユーザが指定した区間ごとに閲覧できるため，新しい人物が現れたり，人物関係が変化したりすることも確認できる．加えて，表示するノードを選択したり，指定した人物が登場する本文に素早くアクセスしたりすることも可能である．我々のインタフェースを活用して，登場人物の関係を視覚的に確認することで，続編を読むために最初から読み直したり，確認のために 1 ページごとに探索したりすることなく，内容を効率良く想起できる．

本稿では，第 2 章で本インタフェースを使用する想定場面を述べ，物語の内容想起支援に必要な情報を整理する．第 3 章で関連研究を挙げ，既存研究との比較を行う．第 4 章にインタフェースの概要を例とともに順を追って示

†1 現在，電気通信大学
Presently with The University of Electro-Communications
†2 現在，JST CREST
a) tanaka@onailab.com
b) m.o@acm.org
c) rikioonai@gmail.com
*1 http://syosetu.com/

し，第 5 章でインタフェースの詳細を述べる．第 6 章でモジュールの概要について述べ，第 7 章でモジュールの詳細を述べる．第 8 章でインタフェースの出力例を示し，第 9 章でまとめる．

2. 物語の内容想起支援

ユーザが本インタフェースを利用する場面のひとつとして，以下のような場面を想定している．

ユーザは村上春樹の 1Q84 シリーズの 1 巻 [*2] を過去に読んだことがあるが，その内容を忘れてしまった．今回 1Q84 シリーズの 2 巻を読むにあたり，1 巻の内容を想起したいが，最初から読み返すのは面倒である．インターネットで検索して調べるにしても，出版社のホームページ [*3] には結末まで書かれておらず，Wikipedia[*4] などではこれから読む 2 巻の内容まで知ってしまう恐れがある．そこでユーザは本インタフェースを用いて，2 巻を読むにあたって必要な内容を効率良く想起する．

内容想起支援に必要な情報は，物語の筋書き，登場人物の設定，登場人物の関係，舞台背景，伏線，重要な台詞など多数考えられる．ここで，物語の筋書きを確認するための関連技術として，文章の自動要約がある．Mihalcea らは自動要約を今まで使われてきた要約技術を改良して物語に応用した [2]．しかし，作成される要約の文章量が大きく，自動で得られた要約そのものを読むのに時間がかかるという問題がある．日本語の物語に適用した例としては，談話理解モデルに基づいた重要文抽出による自動要約 [7] がある．これは，対象を 30 文から 50 文程度の短い物語としており，1Q84 シリーズの 1 巻のような長編の物語には適用できない．別のアプローチとしては，単語に影響区間を付加することで物語の抽出を行う研究 [9] がある．これは，物語から自動で抽出したキーワードとその影響区間を示した表を提示することで，ユーザが過去に読んだ物語の一場面を想起できることを示した．しかし，実験では「はなさかじいさん」という短い物語を使用しており，長編の物語ではユーザが確認するキーワード数が膨大になるといった問題が生じる．

本研究では，内容想起支援に特に重要な情報を「登場人物の関係」と考えた．ここで理由を 3 つ挙げる．1 つ目は，先に述べた 1Q84 の想定場面において，登場人物の情報が必要となるためである．通常，1 巻の登場人物は 2 巻でも引き続き登場し，「主人公の名前」，「主人公と友好関係にある人物」，「主人公と敵対関係にある人物」などを把握することで，2 巻を円滑に読むことが可能となる．実際，市販されている物語の一部では，巻頭に登場人物の名前や年齢，所属，他の人物との関係性などを整理して記載している．

2 つ目は，登場人物の関係の変化が物語の筋書きに密接に関係しているためである．日本のおとぎ話のひとつである「桃太郎」を例にすると，桃太郎とサルは最初から仲間ではないし，桃太郎は生まれたときから鬼と戦っていたわけでもない．桃太郎が成長し，道中でイヌ，サル，キジを仲間にし，最後に鬼と戦うことになる．このことからも，物語が進行していくと共に，登場人物の関係も変化していることが分かる．3 つ目は，登場人物は物語のジャンルによらず重要な情報となるためである．物語のジャンルとしては，恋愛，ミステリー，SF などが考えられる．恋愛においては登場人物の恋愛関係，ミステリーにおいては伏線，SF においては舞台背景など，物語のジャンルごとに重要な情報は異なる．しかし，特定のジャンルに限った情報に注目すると，インタフェースの利用範囲が限定的になるという問題がある．そこで，ジャンルによらずに登場する「人物」に注目して人物関係を視覚化する．

本インタフェースでは，「登場人物の関係」を視覚化することで，効率の良い内容想起支援を目指す．システムは，物語のテキストから自動的に人物候補を抽出する．ユーザは，抽出された人物候補をインタフェース上で確認し，マウスによる簡単な操作で修正する．次に，システムが自動的に人物間の関連度の強さを計算し，インタフェース上に登場人物の関係図を表示する．ユーザは，登場人物の関係図を任意の区間ごとに見たり，人物を含む文章を見たりすることで，登場人物の関係を確認し，過去に読んだ物語を想起する．

本章では，想定場面のひとつとしてシリーズ物の続編を読むときを取り上げたが，本インタフェースは読書を中断したときにも活用できる．通勤時間などを活用した読書，トイレ休憩や電話対応などで，読書を中断せざるを得なくなる場面は多い．読書を再開する際，入力として読んだところまでのテキストデータを与えることで，内容を効率良く確認できる．さらに，本インタフェースは登場人物の関係図を出力するため，物語を読んでいる途中で内容が分からなかったときにも利用できる．例えば，推理小説で「犯人は○○だった」という表現を読んだとき，「○○」がどこで登場したどんな人物か分からないと物語を楽しめない．また，三国志のように，字面や発音が類似する登場人物が複数登場する場合，登場人物の情報の整理が面倒である．登場人物の関係図を閲覧することで，過去に読んだページをひとつずつ探索することなく，登場人物の情報の整理が可能となる．

3. 関連研究

単語の出現頻度や単語間の関係を視覚化することで，内容理解を支援するシステムは多数存在する．Viégas らは，単語の出現頻度で文字サイズに重み付けし，ランダムに配置することで視覚化した [4]．しかし，単語間の関係が分か

[*2] 村上春樹：1Q84 BOOK 1，新潮社 (2009)
[*3] http://www.shinchosha.co.jp/book/353422/
[*4] http://ja.wikipedia.org/wiki/1Q84

らないという問題がある．Collins らは，単語の出現回数や IS-A 関係を放射状に表示することで視覚的な要約を提供した [1]．また，van Ham らは文章を高速に解析し，ユーザの指定した A of B や A and B といった関係を有向グラフで視覚化した [3]．これらの手法は単語間の関係を視覚化しているものの，前者は WordNet の IS-A 関係を利用しているため固有名詞が多く登場する物語に適用できない，後者は「X of Y」といった正規表現で抽出しているため, 日本語の物語にそのまま適用できないという問題がある．

本インタフェースでは，「登場人物の関係」を視覚化するにあたり，物語のテキストから自動的に登場人物を抽出する．米田らは，頻度と述語情報を利用して未知の人物にも対応可能な抽出手法を提案した [11]．この手法は，短編小説 30 作品を対象にした実験において，適合率 60.3%，再現率 91.9%，F 値 71.5%という良い性能を得た．しかし，登場人物を抽出する対象から会話文を除外するため，会話文のみに登場するニックネームなどを抽出できない．また，この手法を利用する場合，学習データから自動で抽出した数百個の人物候補が人物かどうかをあらかじめ手動で判断しておく必要がある．そこで本研究では，日本語解析ツール MeCab[*5] と CaboCha[*6] の固有表現解析の結果に，頻度情報を考慮した tf-idf を活用した登場人物の抽出結果を加えることで，登場人物の抽出を容易に実現した．人物以外が誤って抽出されても，ユーザが簡単なマウス操作で「関係図に表示しない人物」に指定することで，登場人物の関係図から人物以外の単語を除外できる．登場人物の関係を視覚化する研究としては，共起頻度に基いて関係図を描画する研究 [10] や登場人物の会話に着目して関係図を描画する研究 [6] が存在する．これらの手法は，人物関係を計算する際，「ふかえり」に対する「深田絵里子」といった同一人物を表す別表現を考慮できない．そこで本研究では，同一人物を表す別表現を「言い換え表現」と定義し，この言い換え表現をユーザのマウスによる簡単な操作のみで指定できるようにした．これにより，人物関係の計算に言い換え表現を反映させた．また，本インタフェースでは入力した物語に対して関係図を 1 つ出力するのではなく，ユーザが指定した区間の関係図を閲覧できる．これにより，新しい登場人物が現れたり, 人物関係が変化したりすることも確認可能となった．

4. インタフェース概要

ここでは，本インタフェースを使用する流れを 1Q84 シリーズの 1 巻を例に順を追って示す．まず，ユーザは過去に読んだ 1Q84 シリーズの 1 巻のテキストデータを用意する．通常，物語は複数の章に分けられており，1Q84 シリーズの 1 巻は全部で 24 章からなる．ユーザはテキストデータを入力として与える前に，章と章の境目にあらかじめ 2 個以上の改行を挿入する．そして，インタフェースの入力画面 (**図 1**) から，物語のタイトルとテキストデータを入力する．テキストデータはテキストボックスに貼り付けても，「ファイルを選択」でファイルを指定しても良い．必要な情報の入力を終えて「送信」を押すと，人物情報の編集画面に遷移する (**図 2**)．ここでは，自動で抽出された人物が，「関係図に表示する人物」，「言い換え表現」，「関係図に表示しない人物」の 3 つに分類されて配置されている．ユーザはこれらの情報を見て，分類を変更したい人物があればマウスによる簡単な操作で適当な位置に再配置する．

人物情報の編集を終えて画面下部にある「送信」を押すと，想起支援画面に遷移する (**図 3**)．想起支援画面では，ユーザが指定した区間の登場人物の関係図を閲覧することができる．また，登場人物の関係図のノードをダブルク

図 1 入力画面

図 2 人物情報の編集画面

[*5] http://mecab.sourceforge.net/
[*6] http://code.google.com/p/cabocha/

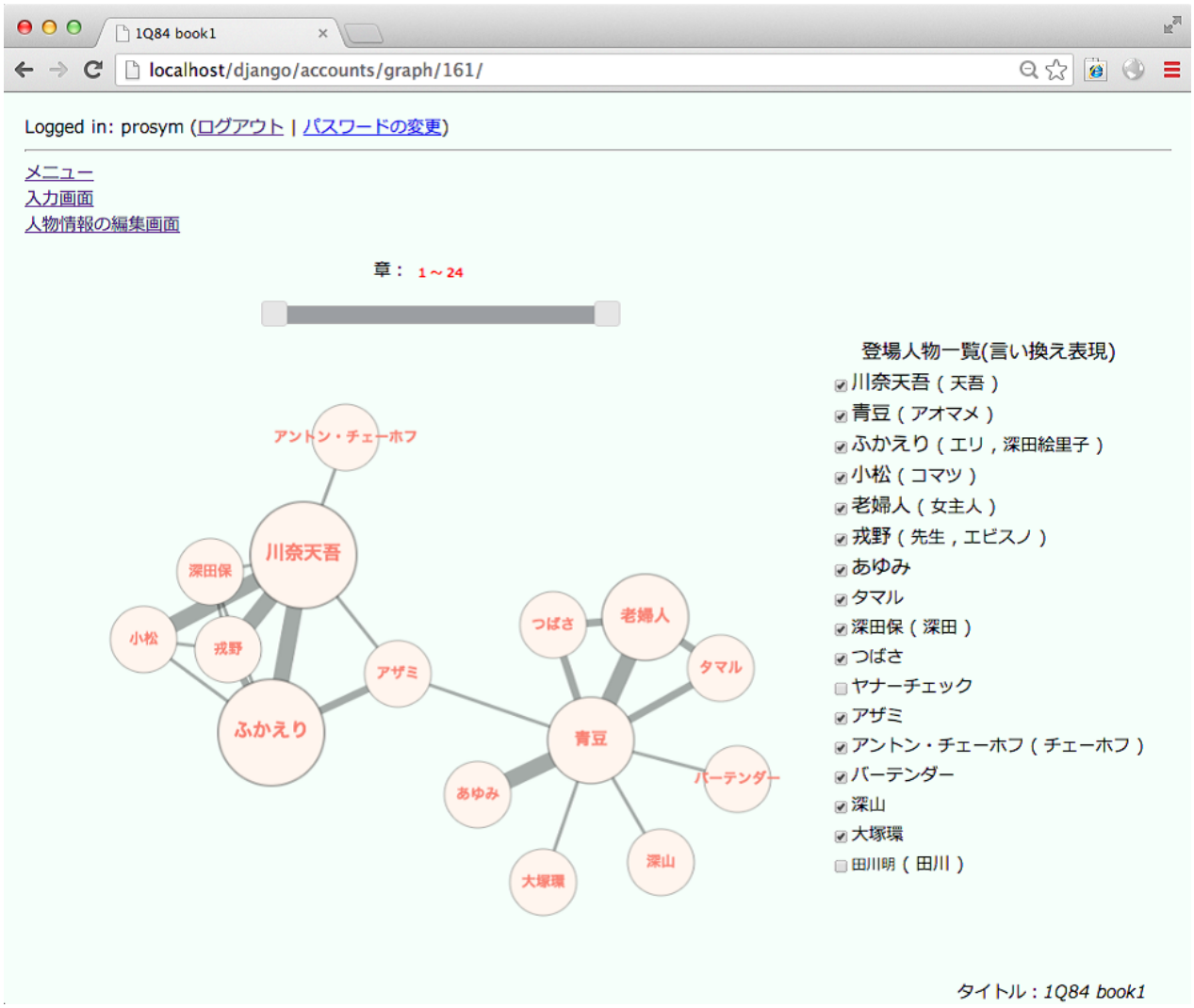

図 3 想起支援画面

リックすると，新しいウインドウが開き，ノードに書かれていた人物名の入った本文を章ごとに閲覧できる．ユーザはこれらの情報を閲覧することで，1Q84 シリーズの 1 巻の登場人物や人物関係の確認を行い，2 巻を読むにあたって必要な内容を得る．

5. インタフェース詳細

ここでは，前章に引き続き村上春樹の 1Q84 シリーズの 1 巻を例として，人物情報の編集画面および想起支援画面の詳細を述べる．

5.1 人物情報の編集画面

人物情報の編集画面 (図 2) において，左から 2 列目が関係図に表示する人物，左から 3 列目が言い換え表現，左から 5 列目が関係図に表示しない人物である．ユーザは，表中の人物の書かれた要素をドラッグして適当な位置にドロップすることで，自動で行われた分類を編集する．関係図に表示する人物と関係図に表示しない人物の移動は，人

図 4 人物情報の編集画面 (編集中の様子)

物の書かれた要素をダブルクリックすることでも可能である．図 4 は，関係図に表示しない人物として分類されていた「深田絵里子」を「ふかえり」の言い換え表現に追加している様子である．この人物情報の編集を行うことで，文字列のマッチングでは抽出できない「ふかえり」に対する「深田絵里子」といった言い換え表現の追加が可能となる．また，「空気さなぎ」，「ショルダーバッグ」などの人物でない単語が人物として抽出された場合，これらの人物を関係図に表示しない人物に指定することで，登場人物の関係図に人物以外の単語が含まれることを防止できる．

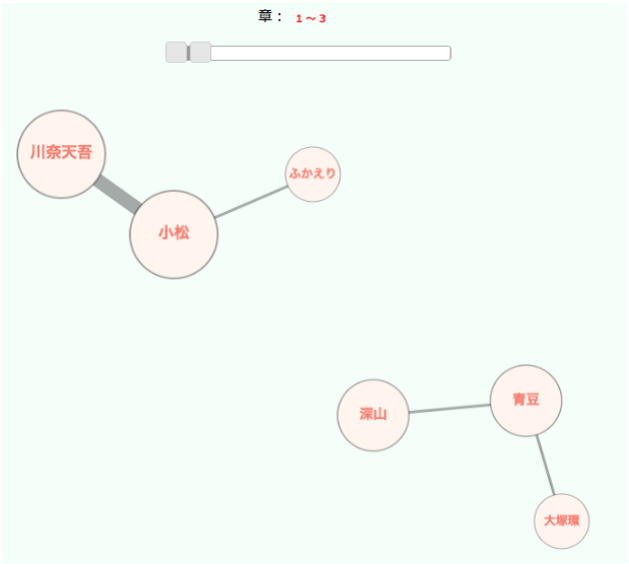

図 5 1 章から 3 章における登場人物の関係図

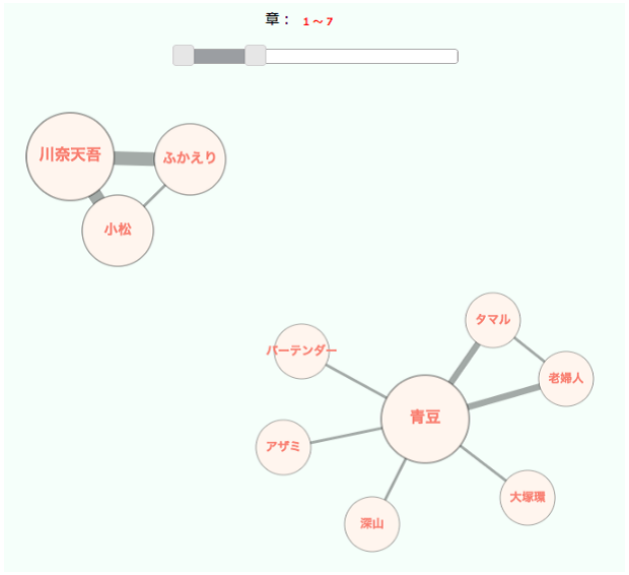

図 6 1 章から 7 章における登場人物の関係図

5.2 想起支援画面

想起支援画面 (図 3) において，画面の上部にあるのが関係図を表示する章の区間を指定するバー，中央にあるのが登場人物の関係図，右部にあるのが登場人物一覧である．区間を指定するバーは，登場人物の関係図を表示する最初の章と最後の章を指定できる．ユーザは，バーを動かすことで区間を変更し，その結果，登場人物の関係図が変化する．**図 5** は 1 章から 3 章までの登場人物の関係図である．ここで，バーを動かして区間を 1 章から 7 章に変更することで，**図 6** のように 1 章から 7 章までの登場人物の関係を閲覧可能となる．これにより，新たに「タマル」が登場したり，「川奈天吾」と「ふかえり」につながりができたりすることが確認できる．これは，物語の内容とも一致している．実際，タマルは 7 章で初めて名前が登場し，川奈天吾とふかえりは 4 章で初めて会っている．

図 7 人物「ふかえり」を含む文を閲覧している様子

登場人物の関係図の各ノードはドラッグして移動することができる．これにより，ノード間を結ぶエッジがノードの後ろに隠れてしまっても閲覧可能になる．図 3 において，ノードをダブルクリックすると**図 7** のような新しいウインドウが開き，そのノードに書かれていた人物の入った文を章ごとに閲覧できる．物語の内容を想起するために重要な単語のひとつである人物の入った文を閲覧することで，内容を効率良く確認できる．実際，短編小説の要約を効率良く作成するために，章ごとに tf-idf の高い単語を含む本文を閲覧することが有効であると確認されている [8]．人物の入った文を閲覧する画面では，ダブルクリックしたノードに書かれていた人物は赤色，その言い換え表現はピンク色，それ以外の人物は青色で強調される．また，人物を 2 名以上含む文は，全体が太字で強調される．表示する章の切り替えは，図 7 の上部にあるバーを動かすか，バーの下にある章番号の書かれたタブをクリックすることで行う．図 7 は「ふかえり」を含む文を閲覧している様子である．ここで，全体が太字で強調されている文を以下に示す．

- しかしそれはそれとして，あの戎野先生がふかえりの保護者になっているとは思ってもみなかったな．
- 「小松さんの能力を疑うわけじゃありませんが，ふかえりはそこらへんの普通の女の子とは違います．

この 2 文のうち，1 つ目の文のみで，「ふかえり」の保護者が「戎野」であることが確認できる．

また，表示するノードを任意に選択することが可能である．表示するノードの選択は，登場人物一覧にあるチェックボックスを切り替えることで可能である．例えば，「川奈天吾 (天吾)」と記述されている左側のチェックボックスをオフにすると，登場人物の関係図から「川奈天吾」と書かれているノードを除去する (**図 8**)．なお，登場人物一覧における括弧内は言い換え表現である．この機能を使用することで，物語の主要でない人物のノードを除去したり，ある人物が存在しないときの関係図を見たりすることができる．

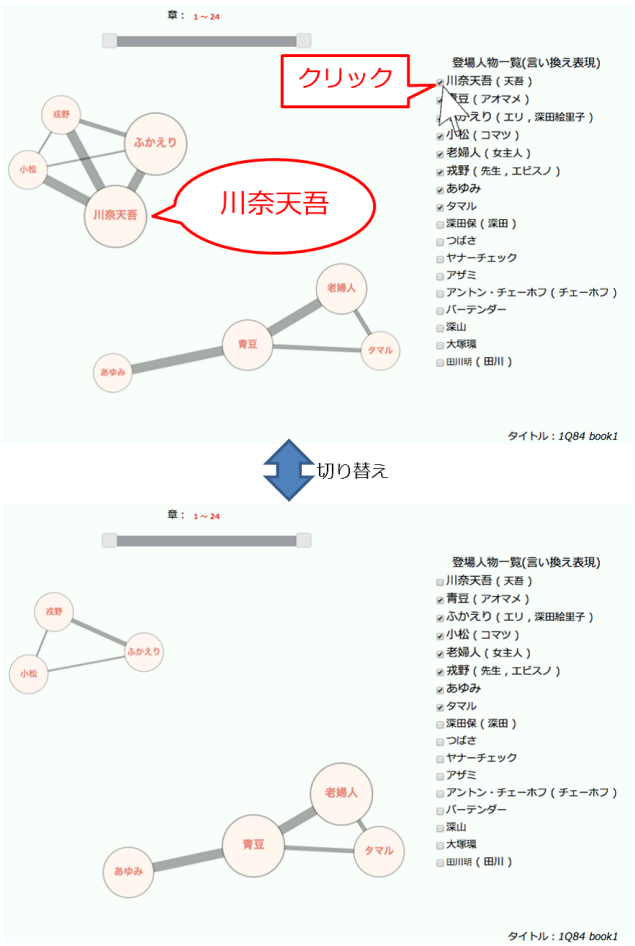

図 8　表示するノードを切り替えている様子

6. モジュール概要

モジュール概要を画面遷移とともに図 9 に示す．モジュールは大きく分けて，人物の抽出，関連度の計算，想起支援画面の出力の 3 つである．人物の抽出では，テキストデータを解析して抽出した人物を，関係図に表示する人物，言い換え表現，関係図に表示しない人物の 3 つに分類する．関連度の計算では，ユーザが修正した人物の分類を受け取り，係り受け解析からルールベースで抽出された主語，目的語，述語とマッチングをとることで，人物間の関連度を計算する．想起支援画面の出力では，関連度の計算結果をもとに物語の内容を視覚化する．想起支援画面において，ユーザが人物の分類を再修正する場合は人物情報の編集画面に戻る．また，ユーザが関係図を表示する章の区間や関係図に表示する人物を変更した場合，関連度の計算を再び行い，想起支援画面を更新する．

7. モジュール詳細

ここでは，人物の抽出，関連度の計算，想起支援画面の出力について詳細を述べる．

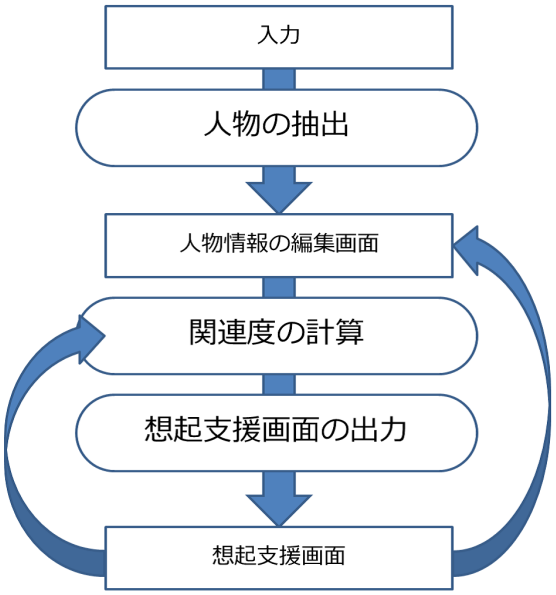

図 9　モジュール概要図

7.1 人物の抽出

テキストデータからの人物の抽出は，2 段階に分けて行う．まず，日本語解析ツール MeCab と CaboCha の固有表現解析を利用する．固有表現解析による人物の判定では，一部人物でない単語も人物として判定されてしまうが，今回は人物と判定された単語すべてを人物候補とする．だたし，連続して人物と判定された単語については，1 つの人物候補として扱うことにする．例えば，「大塚環は二塁手でチームの要で，キャプテンもつとめていた．」という一文では「大塚」と「環」とが人物と判定されるが，これを「大塚環」として扱う．また，「さん」,「様」といった頻出の敬称がついた人物については，敬称を外した状態と同一視する．例えば，「青豆さん」については，敬称「さん」を外した「青豆」と同一視した． 次に，固有表現解析では抽出できない人物を tf-idf を活用して抽出する．固有表現解析では，既存の辞書には存在しない人物「ふかえり」は形態素解析がうまくいかず，人物として抽出できない．また，固有名詞が登場せず「バーテンダー」や「老婦人」などと表現されている場合も対応できない．そこで，テキストデータ n 文字ごとに切り出して tf-idf を計算し，その上位語を取り出す．tf-idf の高い文字列が必ずしも人物とは限らないが，物語文においては tf-idf が高い文字列が人物であることが多い．人物以外が誤って抽出されても，ユーザが人物情報の編集画面で操作することで，登場人物の関係図に人物以外の単語が含まれることを防止できる．

ここで，tf-idf の計算方法を述べる．文字列 t の出現頻度である $tf(t)$ は，文字列 t の出現回数を全文字列数で割ったものである．また，文書頻度の逆数である $idf(t)$ は物語の章の数を 1.1 倍した値を，文字列 t が 1 回以上出現する章の

数で割った値の自然対数である．物語の章の数を 1.1 倍としているのは，すべての章に出現する文字列の $idf(t)$ が 0 となることを防ぐために設定した．tf-idf は，$tf(t)$ と $idf(t)$ の積に，切り出した文字列の長さに依存する重み $w(n)$ をかけたものである (式 1，2)．重み $w(n)$ は，より長い文字列を優先して抽出するために，文字列の長さが長いほど値が大きくなるように設定した．これにより，「ふかえり」の部分文字列である「ふかえ」や「かえり」の tf-idf を相対的に小さくする．

$$tfidf(t,n) = w(n) \times tf(t) \times idf(t) \quad (1)$$

$$w(n) = 0.1(n+9) \quad (2)$$

今回は物語中の人物の大半が 2 文字から 8 文字で表されると仮定して，切り出す文字列 n を 2 から 8 までの整数とした．そして，tf-idf を計算して結果を降順に並び替え，上位の文字列を人物候補とする．ただし，人物の可能性が低い文字列は人物候補から除外する．人物の可能性が低い文字列とは，句読点といった記号を含むもの，先頭が促音のもの，末尾が「は」,「が」,「の」のもの，平仮名 2 文字のみで構成されるものなどである．さらに，上位に存在する文字列の部分文字列となっている場合についても人物候補から除外する．ここで，人物候補とする文字列の数が多いと人物以外の文字列が多数抽出され，人物候補とする文字列の数が少ないと主要な登場人物を抽出できない．本研究では，1Q84 シリーズの 1 巻の他に，ジャンルの異なる数作品で予備実験を行った結果を踏まえ，上位 20 個を人物候補として固有表現解析で抽出された結果に追加した．**表 1** は，1Q84 シリーズの 1 巻について計算した結果のうち，tf-idf の高い上位 15 個である．人物として判定された「天吾」,「青豆」,「ふかえり」,「老婦人」以外の文字列については，人物候補から除外した理由を記述した．例えば，「ふかえ」,「かえり」,「えり」については，より tf-idf の高い「ふかえり」の部分文字列のため除外する．**表 2** は，最終的に抽出した人物 20 個である．tf-idf を活用することで，固有表現解析では抽出できなかった「ふかえり」,「老婦人」,「女主人」,「先生」,「ヤナーチェック」を人物候補として抽出できた．一方で，作中に登場するグループを表す「さきがけ」,「教団」などが誤って人物候補として抽出された．

続けて，抽出した人物を関係図に表示する人物，言い換え表現，関係図に表示しない人物の 3 つに分類する．固有表現解析で抽出した人物候補については，登場回数が一定数を超えるものを関係図に表示する人物とし，tf-idf を活用して抽出した人物候補については，そのすべてを関係図に表示する人物とする．ただし，2 つ方法で分類先が異なる場合は、tf-idf を活用して抽出した結果を優先する．言い換え表現については，文字列を比較することで行う．ある人物に対して，その人物を部分文字列とする人物が 1 つのみ存在する場合，文字数の長いものを関係図に表示する人物，短いものを言い換え表現とする．例えば，人物として「川奈天吾」と「天吾」が抽出されたとき，もし，「天吾」という文字列を含む人物が「川奈天吾」以外に存在しなければ，「川奈天吾」を関係図に表示する人物，「天吾」を言い換え表現とする．

表 1 計算結果 (上位 15 個)

文字列	tf-idf	除外理由
天吾	0.00252	-
青豆	0.00230	-
青豆は	0.00165	末尾が「は」
天吾は	0.00161	末尾が「は」
た。	0.00157	記号を含む
豆は	0.00152	末尾が「は」
吾は	0.00147	末尾が「は」
った	0.00124	先頭が促音
ふかえり	0.00116	-
ふかえ	0.00107	上位に存在する語の部分文字列
かえり	0.00107	上位に存在する語の部分文字列
えり	0.00098	上位に存在する語の部分文字列
てい	0.00095	平仮名 2 文字のみで構成
老婦人	0.00093	-
ない	0.00091	平仮名 2 文字のみで構成

表 2 抽出した人物 (上位 20 個)

文字列	tf-idf	固有表現解析での抽出
天吾	0.00252	可
青豆	0.00230	可
ふかえり	0.00116	不可
老婦人	0.00093	不可
あゆみ	0.00070	可
小松	0.00066	可
さきがけ	0.00056	(人物でない)
教団	0.00040	(人物でない)
深田	0.00040	可
タマル	0.00038	可
空気さなぎ	0.00038	(人物でない)
女主人	0.00036	不可
ギリヤーク	0.00036	(人物でない)
リトル・ピープル	0.00032	(人物でない)
先生	0.00030	不可
ヤナーチェック	0.00029	不可
ギリヤーク人	0.00027	(人物でない)
日曜日	0.00027	(人物でない)
書き直	0.00027	(人物でない)
集金	0.00027	(人物でない)

7.2 関連度の計算

抽出した人物間の関連度の計算を章ごとに行う．関連度を計算するにあたって，入力として受け取ったテキストを文ごとに分割する．一般的に，文末は句点，閉じ括弧，感嘆

符，疑問符のいずれかである．そこで，これらの文字を文末の根拠とする．次に，分割した文ごとに CaboCha で構文解析を行い，ルールベースで主語，目的語，述語を抽出する．本研究では，「人物 A が人物 B に○○した.」や「人物 A は人物 B を××した.」のような文を多く含むほど，人物 A と人物 B の関連度が強いと考えた．例えば，「天吾はふかえりを新宿駅まで送った.」という文からは，「天吾」と「ふかえり」の関係が抽出できる．しかし，人物を 2 名含む典型的な文は，物語の中で多く登場するわけではない．実際，関連度の計算を上記の情報のみを使用して行ったところ，登場回数の多い人物の関係は抽出できたが，登場回数の少ない人物の関係は抽出できなかった．そこで，人物 A と人物 B が同一場面で共起する回数が多いときについても，人物 A と人物 B の関連度が強いと考えた．同一場面の範囲については，前後の文，形式段落，意味段落などが考えられ，馬場ら [10] は，「登場人物の入れ替わり」，「場所の変化」，「時間の経過」を指標として人手で分割した意味段落を同一場面として関連度を計算した．本研究では，ユーザによる手作業を最小限に留めるために，同一場面として前後の文を利用する．ここで前後の文とは，ある文に注目したときの 1 つ前の文および 1 つ後ろの文のことである．これにより，会話文と地の文の両方を読まないと分からない場合やひとつの出来事を複数の文に分割して説明している場合についても関連度に反映可能となる．例えば，「『ねえ、タマルさん、最近月を見たことはある？』と青豆は尋ねた.」からは，会話文で登場する「タマル」と地の文で登場する「青豆」の関係が抽出できる．また，「青豆もそれを口に入れた．二人がそうするのを見届けてから，つばさも同じようにそれを食べた.」からは，連続した 2 つの文に登場する「青豆」と「つばさ」の関係が抽出できる．

ここで，ある章に全部で N 個の文が存在するとき，その章における人物 A と人物 B の関連度 $R(A,B)$ の計算方法を述べる．$R(A,B)$ は，k 番目の文における人物 A から見た人物 B との関係 $r_k(A,B)$ の 1 番目から N 番目の文までの総和と，k 番目の文における人物 B から見た人物 A との関係 $r_k(B,A)$ の 1 番目から N 番目の文までの総和で表される (式 3)．

$$R(A,B) = \sum_{k=1}^{N} (r_k(A,B) + r_k(B,A)) \tag{3}$$

k 番目の文における人物 A から見た人物 B との関係 $r_k(A,B)$ は，k 番目の文の主語 ($subject_k$) に人物 A を表す表現が存在するとき，人物 B を表す表現の存在状態に依存する関数 $E_k(B)$ を用いて計算される (式 4)．ここで，$E_k(B)$ は k 番目の文の主語，述語，目的語のいずれかに人物 B を表す表現が存在すれば 1，存在しなければ 0 とする．ただし，0 番目の文，$N+1$ 番目の文は存在しないので，$E_0(B)$，$E_{N+1}(B)$ はそれぞれ 0 と定める．今回は同一場面で人物が共起する回数に比較して，人物を 2 名含む典型的な文の数が少ないことを考慮し，$E_k(B)$ に $E_{k-1}(B)$ と $E_{k+1}(B)$ の 3 倍の重み付けをした．また，各人物を表す表現には，その言い換え表現も含めることにする．例えば，人物「ふかえり」を表す表現としては，「ふかえり」の他に，その言い換え表現である「深田絵里子」も含める．

$$r_k(A,B) = \begin{cases} E_{k-1}(B) + 3E_k(B) + E_{k+1}(B) & (\text{if } A \text{ in } subject_k) \\ 0 & (\text{otherwise}) \end{cases} \tag{4}$$

関連度の計算は，関係図に表示する人物すべての組み合わせについて章ごとに行い，計算結果を保存する．また，関連度とは別に，各人物を表す表現の登場回数も保存する．複数の章にまたがった関連度，例えば 1 章から 3 章における人物 A と人物 B の関連度は，各章で求めた関連度 $R(A,B)$ の和を用いる．

7.3 想起支援画面の出力

登場人物の関係図の描画は，arbor.js[*7] を拡張することで行った．arbor.js は，JavaScript ライブラリである jQuery に依存するオープンソースのグラフ描画ライブラリで，ノードの位置決定やノードに対する操作が可能である．ノードの位置決定には力学モデルを用い，エッジの長さを可能な限り均一にし，エッジ同士が可能な限り交差しないようにしている．登場人物の関係図では，人物 A が登場しているときに人物 A と書かれたノードが描画され，他の人物との関連度が 1 以上あるときにそのノードとエッジで結ばれる．ここで，ノードの大きさは登場人物の関連度の総和に応じて 3 段階に分けた．つまり，多くの人物と関連を持っていたり，関連度の強い人物がいたりすると大きく表示される．また，登場人物一覧の文字サイズは，その人物を表す表現の登場回数に応じて 3 段階に分けた．このとき，登場回数の多い人物ほど大きく表示されるようにする．なお，登場回数の少ない人物は，主要でない人物の可能性が高いため，初期状態ではグラフに表示しないことにした．

8. インタフェースの出力例

ここまでは，1Q84 シリーズの 1 巻を例に挙げた．本章では，他の作品の例として，東野圭吾「マスカレード・ホテル」[*8] のテキストデータを入力として与えた場合を掲載する．マスカレード・ホテルでは，意味段落ごとに通し番号が付与されており，今回は通し番号を章と章の境目とする．**図 10** は，東野圭吾「マスカレード・ホテル」を入力として与え，本論文執筆者の一人が人物情報の編集画面で 2 分程度操作したときに得られる結果である．右の登場人物一覧では，人物として「ｘ４」が抽出されている．「ｘ４」は

*7 http://arborjs.org/
*8 東野圭吾：マスカレード・ホテル，集英社 (2011)

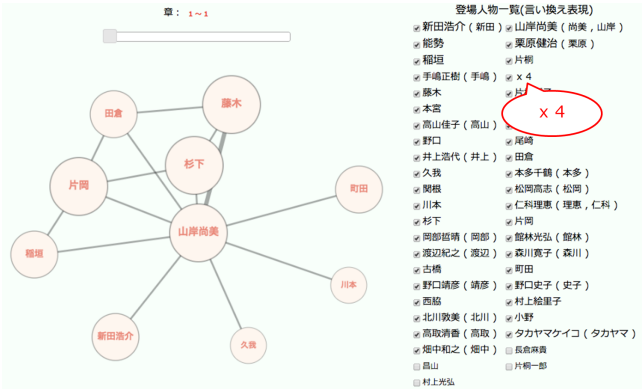

図 10 東野圭吾「マスカレード・ホテル」を入力とした場合

作中の事件の容疑者であり，重要な人物であるが，固有表現解析では抽出できない．提案した tf-idf を活用した抽出により，重要な人物「ｘ４」を抽出できたことが確認できる．関係図に表示する登場人物が多くなったときに，ノードが重なって読み取りにくい点については，今後の課題である．

9. おわりに

本研究では，「内容想起支援のためのインタフェース」として「登場人物の関係」の理解を支援するシステムを開発した．登場人物の関係を視覚化するにあたって，人物の抽出手法および人物間の関連度の計算手法を提案し，実装した．人物の抽出手法においては，既存の固有表現解析の結果に tf-idf を活用した抽出結果を加えることで，手作業で学習データを用意することなく，未知の人物や固有名詞以外で表現される人物の抽出が可能となった．人物間の関連度の計算手法においては，人物を 2 名含む典型的な文と前後の文の共起情報を利用した．自動で抽出・分類した人物情報をユーザがマウスによる簡単な操作で修正することで，「ふかえり」に対する「深田絵里子」といった単純な文字列比較では対応できない言い換え表現を関連度の計算に反映させることが可能となった．ユーザは，マウスによる簡単な操作のみで，物語の登場人物の関係を任意の区間ごとに閲覧し，新しい人物が現れたり，人物関係が変化したりすることを確認可能である．本インタフェースを使用することで，過去に読んだ本を最初から読み返すことなく内容を想起できる．

今後の課題としては，登場人物の関係の詳細の抽出である．人物 A と人物 B の関連度のみでなく，「友人」，「仕事仲間」といった人間関係を抽出してユーザに提示することで，より効率の良い内容想起支援を目指す．また，物語の内容想起支援における本インタフェースの有効性をユーザスタディを通して確認し，ユーザからのフィードバックをもとに，物語を想起する上で必要な情報とその提示方法について，さらなる検討を行う．

参考文献

[1] Collins, C., Carpendale, S. and Penn., G.: Docuburst: Visualizing document content using language structure, *Computer Graphics Forum*, Vol. 28, No. 3, pp. 1039–1046 (2009).

[2] Mihalcea, R. and Ceylan, H.: Explorations in Automatic Book Summarization., *EMNLP-CoNLL*, pp. 380–389 (2007).

[3] van Ham, F., Wattenberg, M. and Viégas, F.: Mapping text with phrase nets, *Visualization and Computer Graphics, IEEE Transactions on*, Vol. 15, No. 6, pp. 1169–1176 (2009).

[4] Viégas, F., Wattenberg, M. and Feinbergn, J.: Participatory visualization with wordle, *Visualization and Computer Graphics, IEEE Transactions on*, Vol. 15, No. 6, pp. 1137–1144 (2009).

[5] 伊藤雄一， 縣啓治，高嶋和毅：ヨミログ：読書ログによる個人に応じた読了時間推定システム，情報処理学会シンポジウムインタラクション 2012 論文集，pp. 129–136 (2012).

[6] 神代大輔，高村大也， 奥村学：物語テキストにおけるキャラクタ関係図自動構築，言語処理学会第 14 回年次大会発表論文集，pp. 380–383 (2008).

[7] 村上聡，榎津秀次，古宮誠一：物語自動要約手法の提案：談話理解モデルに基づいた重要文抽出，電子情報通信学会技術研究報告. KBSE, 知能ソフトウェア工学， Vol. 104, No. 724, pp. 49–54 (2005).

[8] 田中翔太郎， 岡部誠，尾内理紀夫：物語の要約を支援するインタフェース， *WISS2011*, pp. 30–35 (2011).

[9] 藤井崇介，土井晃一郎，山本章博：単語の出現区間推定を利用した物語構造の抽出，電子情報通信学会技術研究報告. KBSE, 知能ソフトウェア工学， Vol. 106, No. 473, pp. 43–48 (2007).

[10] 馬場こづえ， 藤井敦：小説テキストを対象とした人物情報の抽出と体系化，言語処理学会第 13 回年次大会発表論文集，pp. 574–577 (2007).

[11] 米田崇明，篠崎隆宏，堀内靖雄， 黒岩眞：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，pp. 855–858 (2012).