

物語テキストから進行状況に応じて登場人物の存在状態と関係を推定する手法

Estimation Method of Character's State of Existence and Relationship According to Progress of Storytelling

縣 啓治 伊藤 雄一 高嶋 和毅 北村 喜文 岸野 文郎*

Summary. 本稿では、物語テキストの内容理解を支援するために、物語テキストの進行状況に応じた人物相関図の生成を目的とし、相関図を生成する上で必要となる登場人物の存在状態を判定する手法と、人物間の友好敵対関係を推定する手法を提案する。提案手法では、物語テキストの係り受け解析結果と、あらかじめ生成した死亡表現リストに基づいて存在状態を判定し、それらを物語の進行状況に沿って可視化する。今回は、本のような外観のソフトウェアを実装し、見開き本の左ページに物語テキスト、右ページにユーザが読み進めている時点までの登場人物の存在状態を可視化する。さらに、あらかじめ生成した友好リスト・敵対リストと、係り受け解析結果、人物の関連度に基づいて人物間の友好敵対関係を推定する。実際の作品を対象にし、提案手法における死亡判定について再現率と適合率を用いて評価した結果、それぞれ平均して、84.7%と79.4%の正解率を得た。

1 はじめに

KindleやiPadなどの電子書籍端末が普及しつつあり、有料/無料の電子書籍を販売する体制も整いつつある。電子書籍の場合は、簡単にデータを大量に持ち運べるため、その日の気分に合わせて読む作品を決め、同時に複数の作品を読み進めるといったことも可能となり、読書スタイルが従来に比べ多様化している。このようなスタイルで通勤電車等で細切れに読書をする、読んでいた内容が把握しづらく、特に推理小説のような複雑な人物関係をもつ物語の内容理解は困難である。そこで、物語を簡単におさらいできる機能があれば、無駄な読返し時間を減らせることや、読了をあきらめていた作品に再挑戦する気力をわかせる効果が期待できる。また、もともと読書が苦手だと感じ、読書から遠ざかっている人にとっては、何らかの方法で、内容理解を支援する必要がある。そこで、物語の最も基本かつ重要であり、物語に沿って変化する人物情報をリアルタイムに、かつ、一目で分かる形式で可視化することが、理解支援につながると考えられる。

そこで本稿では、人物の存在状態の把握と、相関図生成の第一歩として人物間の友好敵対関係の把握に着目する。物語テキストの進行状況に応じた相関図を生成する上で、人物の存在状態判定と関係判定が不可欠であると考えられるためである。

提案手法では、人物の存在状態として、登場人物の登場判定と退場判定を行う。ここで、対象を推理

小説とし、退場の原因として対象の死亡状態への遷移を扱う。そして、それらを物語の読む進み具合に応じて、可視化する。さらに、相関図生成には人物の関係の推定が必要であるため、あらかじめ生成した友好リストと敵対リスト、係り受け解析結果、人物の関連度に基づいて、友好敵対関係を推定する。

2 関連研究

テキストの内容理解支援を目的として、テキストを解析し、その内容に応じて画像や動画等のマルチメディアコンテンツを付加することにより、テキスト内容を可視化する様々な研究がなされてきた。

2.1 テキスト情報の可視化による理解支援

テキスト情報を可視化することにより内容理解を支援する研究として、テキストから3次元空間上にシーンを生成する研究がある。Coyneらが提案しているWordsEye[1]では、ユーザが入力した任意のテキストに対して自然言語処理により、キャラクタや物に加え、それらの位置関係やポーズを抽出し、3次元空間上にテキストで表現されたシーンを生成している。他にも、ブログ記事を対象にし、内容を表す画像の提示による支援が、佐藤らにより提案されている[2]。また、交通事故の様子をアニメーション化し、理解を手助けする研究もある[3]。物語テキストを対象に、内容理解を支援する手法も提案されている。Sumiらが提案しているInteractive e-Hon[4]では、テキストから視覚情報に富んだコンテンツを生成する。主語や動詞、目的語等をテキストから抽出し、データベースから、テキストの流れに沿った動作データを取得し、提示する。また、元のテキストから、親子が会話しながら内容を説明する会話文を作成し、アニメーションと共に再生することで、

Copyright is held by the author(s).

* Keiji Agata, Yuichi Itoh and Kazuki Takashima, 大阪大学 大学院情報科学研究科, Yoshifumi Kitamura, 東北大学 電気通信研究所, Fumio Kishino, 関西学院大学 理工学部

絵本を読んで聞かせているような感覚で内容を理解させることが可能となっている。また、松本らは、小説文を対象に、シーンを表現する場所や効果音に適したマルチメディアコンテンツを動的に付加することによって内容理解を促進するシステム mTeX [5] を提案している。これは、文章の係り受け解析結果と Web 上の辞書を用いることによって、シーンが進行している場所に関する語句を抽出している。さらに、抽出した場所に関する画像を Web 上から取得し、表示領域全体の背景として提示する。

以上の手法では、物語テキストを読む際に臨場感を演出することが可能であるが、物語の流れを簡単に見返すことや、複雑な人物関係を把握することはできない。

2.2 物語テキストにおける人物関係図の構築

物語テキストにおける人物関係図を構築する手法として、神代らは、長い物語を読む際の理解の手助けを目的として、物語テキストから登場人物同士の関係を推定し、人物相関図を自動で構築する手法を提案している [6]。物語テキストから各発話文について話し手と聞き手を同定し、会話内容から、目上目下関係を推定する。例えば、話し手が一人称として「わたくしめ」を使っていれば話し手が相手よりも目下の関係にあると推定する。その結果を基に、ノードとエッジによる無向グラフにより人物相関図を出力する。このシステムは、未読の物語を読むか否かの意志決定の際にも応用できると著者らは述べている。

また、馬場らは、登場人物に基づいて小説テキストをモデル化する手法を提案している [7]。重要人物や人物間の関連度の把握を容易にするため、ノードとエッジを用いて人物相関図を作成している。この相関図では、人物間の関連度が高いほど人物を近くに配置しており、重要人物の判断が可能である。また、重要人物の行動に着目することで、あらすじ生成への応用が可能になる。この相関図から、主要人物と人物間の関連の深さは把握できるが、敵対関係や友好関係等、どのような関連があるかまでは把握できない。

他にも、芳村らは登場人物間の関係を理解するために、人物相関文（人物間の関係を表現する文）を抽出し、人物間のリンクとともに表示することで人物相関図を生成している [8]。しかし、この相関図では、人物相関文を読者が読み直さなければ、関係を把握できないため直感的な理解につながらない。

以上のように、人物相関図を生成する研究は存在するが、物語の進行に沿った可視化や存在状態の判定は実現されていない。

他にも、物語テキストではなく、実在の人物を対象にし、Web 上の情報から、人間関係や関連の強さを求めて、人間関係ネットワークを構築している研

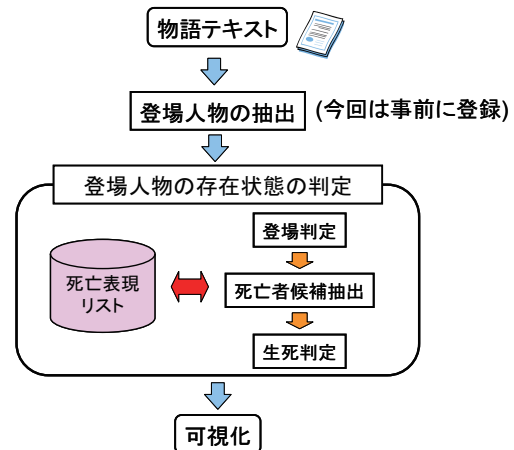


図 1. 提案手法の流れ

究がある [10]。現在では、人々のつながりを可視化し、人物検索を支援する SPYSEE[11] と呼ばれるサービスも存在しており、人物の情報として、その人の相関関係は重要な要素であると言える。

3 人物存在状態の判定

3.1 概要

人物相関図は物語の内容理解の支援に有効であるが、細切れ時間での読書を効果的に支援するためには、読む進み具合に応じた適切な相関図を提示する必要がある。そのため、従来研究のように、物語全体に対する相関図だけでは、内容把握の支援には不十分である。相関図において、最も基本となるのが人物の存在状態（登場したか、退場したか）であると考えられるため、まずはこれを判定し、可視化することに着目する。特に、推理小説では人物関係が複雑で、細切れの読書では人物の状態を把握しきれなくなってしまう恐れがある。そこで、推理小説を対象とし、人物の存在状態を判定する。

提案手法の流れを図 1 に示す。まず、入力された物語テキストから登場人物を抽出する。今回は事前に手入力で人物名を登録しておく。次に、物語テキストの係り受け解析結果とあらかじめ生成した死亡表現リストを基に各登場人物の存在状態を判定する。最後に、得られた人物の存在状態を物語の進行状況に応じて可視化する。

3.2 登場判定

登場人物は自動で抽出できるが、ニックネームに対する同一人物の判定が困難である。また、今回は人物の抽出よりも存在状態判定に着目しているため、人物を事前に登録している。事前に登録しておいた人物のうち誰が登場したのかを判定する手法について述べる。テキスト文に最初に人物名が出てきた時点から、その人物が登場したと判定する。人物名が出てくるということは、その人物がその場面において何らかの関係性があると考えられるためである。

表 1. 死亡表現リストの例

遺体	死	他界	亡くなる	息を引取る
----	---	----	------	-------

各章で登場状態を判定し、人物名が表記されていない章では退場したとみなす。このようにして、各章の主要人物を把握できるようにしている。

3.3 退場判定

物語において、特に連続殺人等の推理小説の場合では、人物の生死状態は重要な意味をもっており、相關図における基本的な要素であると考えられる。そのため、登場判定だけでなく、生死状態の判定も必要である。提案手法では、係り受け解析の結果と死亡表現リストを基に生死状態を判定する。

3.3.1 死亡表現リスト生成

人物が死亡したかどうかを判定するために、青空文庫 [9] に収録されている作品から、登場人物が初めて死亡した際の表現と、既に死亡していると分かる表現を抽出する。さらに、必携類語実用辞典 (Yahoo! 辞書) を利用し、「殺す」「死ぬ」「死体」の類語を調べ、それらを合わせて死亡表現リストとしてまとめた。死亡表現リストに含まれる語句は約 140 語あり、表 1 に示す 5 語はその例である。このように作成した死亡表現リストと係り受け解析の結果を基に、人物の生死状態を判定する。

3.3.2 死亡者候補抽出

まず、人物の生死状態を「生存」「死亡」「死亡候補」の三種類の状態に分類する。そのために、係り受け解析の結果を基にし、登場人物の係り先と係り元の名詞と動詞を抽出する。その際、抽出した語句が死亡表現リストの語句に含まれているかどうかを確認する。登場人物の係り先と係り元の名詞と動詞が、死亡表現リストに含まれていない場合は生存していると判定する。含まれている場合は、その語句が名詞であれば、死亡していると判定し、動詞であれば死亡者候補とする。例えば、図 2 に示すように、「A 男が歩いた」という文では、「A 男」の係り先である「歩いた」が死亡表現リストに含まれていないため、生存していると判定する。「A 男の死体」という文では、「A 男」の係り先である「死体」が名詞で、かつ死亡表現リストに含まれるため、死亡していると判定する。「A 男が自殺した」という文では、「A 男」という人物の係り先の語句である「自殺した」が動詞で、かつ死亡表現リストに含まれるので、死亡候補と判定する。

3.3.3 死亡判定

次に、死亡候補と判定された人物について、生死状態を判定する。人物に続く助詞と、人物の係り先と係り元の文節内の助動詞の意味を手がかりに、生

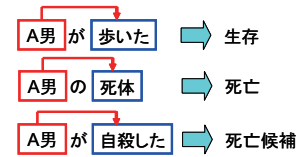


図 2. 生存、死亡例と死亡候補例

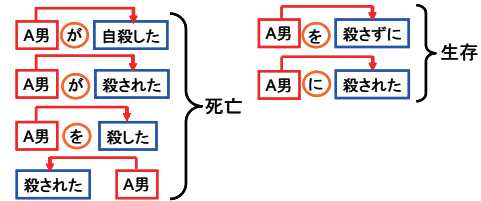


図 3. 生死判定の例

死状態を判定する。判定例を図 3 に示す。打消し、推量の表現がない「A 男が自殺した」のような文の場合は、死亡していると判定する。また、係り先が「殺す」等の他動詞で、人物に続く助詞が「は」「が」「の」「も」のいずれかであり、かつ文節内に受身を表す「れる」がある場合と、人物に続く助詞が「を」である場合のみ、死亡していると判定する。例えば、「A 男が殺された」という文では、助詞が「が」で、かつ係り先が受身を表しているため、死亡していると判定する。「A 男を殺した」という文では、助詞が「を」であるため、死亡していると判定する。「殺された A 男」という文では、係り元が他動詞で、かつ受身を表しているため、死亡していると判定する。

人物の係り先と係り元の文節内に、打消しを表す「ない」「ぬ」や、推量を表す「う」がある場合は、生存していると判定する。例えば、「A 男を殺さずに」という文では、「A 男」の係り先の文節が打消しを表し、かつ他動詞であるため、生存していると判定する。また、「A 男に殺された」という文では、係り先の語句が他動詞で、人物に続く助詞が「に」であり、「は」「が」「の」「も」のいずれでもないため、生存していると判定する。

以上のように、生死判定では、係り受け解析と死亡表現リスト、さらに、人物に続く助詞と、係り先と係り元の文節内の助動詞の意味を利用し、人物の生死状態を判定する。

3.4 実行結果

青空文庫 [9] に収録されている 5 つの作品を対象に、提案手法によって生死の判定を行った。対象作品は、作中で複数の人物が殺害されるものを選択した結果、「有喜世新聞の話」「電気看板の神経」「殺人鬼」「カンカン虫殺人事件」「ネオン横丁殺人事件」とし、各作品に関して再現率と適合率を求めた。ここで、正解死亡箇所の内、提案手法が取得できた割合を再現率とし、提案手法が判定した死亡箇所の内、正解している割合を適合率とした。それぞれ平均で、84.7% と 79.4% の正解率を得た。結果を表 2 に示す。

表 2. 死亡判定の割合 [%]

作品名	再現率	適合率
有喜世新聞の話	80.0 (4/5)	66.6 (4/6)
電気看板の神経	66.6 (2/3)	66.6 (2/3)
殺人鬼	76.9 (10/13)	76.9 (10/13)
カンカン虫殺人事件	100 (2/2)	100 (2/2)
ネオン横丁殺人事件	100 (1/1)	100 (1/1)

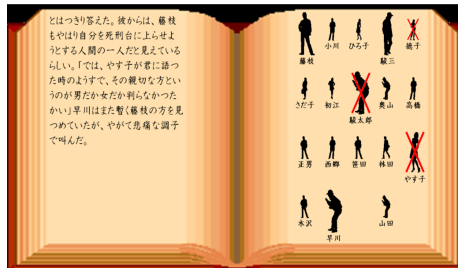


図 4. 人物の存在状態の可視化例

4 人物の存在状態の可視化

前章で判定された結果に基づき、登場人物の存在状態を可視化する手法について述べる．ここでは見開きの本の左ページに物語テキストを表示する．さらに、読み進めている時点での人物の存在状態を逐一確認できるように、右ページには、人物のシルエット画像と人物名を常に表示させておく．シルエット画像は事前に準備し、人物の性別に応じてランダムに提示する．ただし、人物の存在状態は1ページ前の状態を可視化する．これは、表示ページの情報を可視化すると、ページを切り替えた際に、未読の部分の情報を知ってしまう可能性があるためである．

人物が死亡している場合は、その人物のシルエット画像にバツ印を付ける。また、物語中に既に登場しているが、現在表示しているページの章には登場していない人物のシルエット画像は、登場中の人物より小さく表示する。閲覧方法として、1 ページ進む、もしくは、戻るボタンと読返しや流し読みに利用できるスライダバーを実装している。

実行結果の例として、「殺人鬼」という作品を提案手法により解析し、得られた人物の存在状態を可視化した様子を図4に示す。この例から、現在表示しているページまでに、18名の人物が登場し、そのうち3名が死亡していること、現在の章には5名の人物が登場していることが分かる。

また、同様に「電気看板の神経」という作品を提案手法により解析し、得られた人物の存在状態を可視化した例を図5に示す。この作品の8ページ目では、人物が誰も表示されていないが、テキスト文中に「岡安」という人物名が表記されているため、「岡安」が登場していると判定し、9ページ目で、「岡安」のシルエットと名前を表示している。さらに29ページまで進むと、「春ちゃんが惨殺されてしまった」という文から「春江」が死亡していると判定するため、次ページでは「春江」が死亡していると分かるように、「春江」にバツ印をつけている。さらに読み進めると、次々に人物が死亡していき、物語の最後には3名の人物が死亡している様子が分かる。このように、人物の存在状態を読む進み具合に応じて可視化することで、各登場人物が、どの時点で登場し、死亡しているかどうかを把握できる。

5 人物間の友好敵対関係の推定

5.1 概要

前章までの人物の存在状態の可視化により、物語テキストの理解に十分寄与すると考えられるが、さらなる支援のためには、人物の関係を推定し、相関図を生成する必要がある。人物間の関係には、様々な関係が存在するが、恋愛関係や感情等の多くの関係性は、友好敵対関係の一部であると見なせるため、それらが最も基本的な関係であると考えられる。そこで、まず友好敵対関係に着目する。友好敵対関係の推定結果として、二人の人物がどのような関係にあるかを表す語句とその関係にある人物が特定される。そして、友好リスト・敵対リストのどちらに抽出語句が含まれるかを確認することで、二人が友好

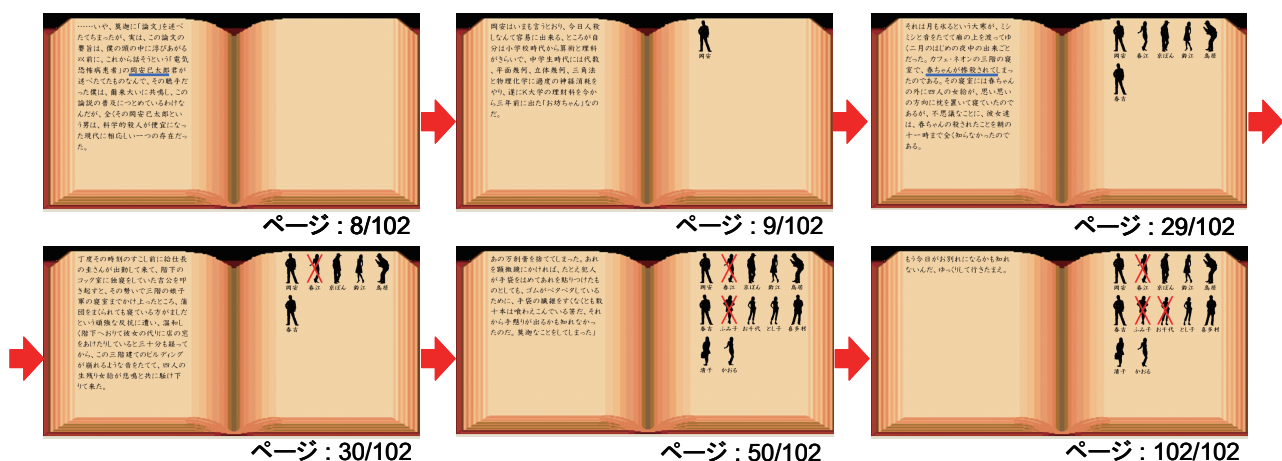


図 5. 「電気看板の神経」の可視化例

表 3. 友好リストの例

親友	仲良し	幼馴染み	惚れる	気がある
----	-----	------	-----	------

表 4. 敵対リストの例

ライバル	敵視	反発	確執	争う
------	----	----	----	----

関係であるか、敵対関係であるかを推定する。

5.2 友好リスト・敵対リストの生成

まず、青空文庫 [9] に収録されている作品から、人物間の友好関係、もしくは、敵対関係にあると分かる表現を人手で抽出し、友好リストと敵対リストとしてまとめた。例えば「圭さんと吉公とはまあ仲のいい方で」という表現では、二人が友好関係にあることが分かる。友好リストには「親友」「仲良し」という語句や、敵対リストには「ライバル」「敵視」という語句が含まれている。友好リスト、敵対リストにはそれぞれ、85 語、48 語を登録している。以降では、友好・敵対リストの語句をキーワードと定義する。表 3 と表 4 にキーワードの例を示す。

5.3 人物の特定

友好・敵対リストと係り受け関係、人物間の関連度を基に、友好敵対関係にある人物を特定し、人物間の友好敵対関係を推定する。

まず、キーワードの係り先と係り元の語句を確認する。それらの語句が人物であれば友好敵対関係にある人物として確定する。確定した人物が二人未満であれば、キーワードの手前に登場する三名とキーワードの後に登場する二名を、友好、もしくは、敵対関係にある候補者とする。候補者をこのように絞ったのは、友好敵対関係にある人物の登場パターンを三つの作品から人手で分析した結果、以下のようなパターン 1~4 に分けられたためである。例文中の下線部は友好、もしくは、敵対関係にある人物を表し、二重下線部はキーワードの語句を表す。

1. キーワード前後で最も近くに登場する人物
(例) 鈴江 は自分の 惚れている 岡安 と
2. キーワードの後で最も近くに登場する二人
(例) 友 に一切の事情を語った。セリヌンティウスは無言で首肯き、メロス をひしと抱きしめた
3. キーワードの手前で最も近くに登場する二人
(例) 鳥居圭三 という三十五にもなる男で...
こいつは、内々春ちゃん に 気があるらしい
4. キーワードの手前で、最も近くに登場する人物と三番目に近い人物
(例) 岡安 も 春ちゃん のことは夢のように忘れちゃったらしく、鈴江 と 肝胆相照らしている。

次に、候補者の中で、現段落までの関連度が最も高い組の人物を友好敵対関係にある人物として確定する。登場回数が多い人物同士は偶然共起している

可能性があるため、Dice 係数を用いて人物の関連度を式 (1) により算出する。

$$\frac{2 \times (\text{人物 A と人物 B の共起段落数})}{(\text{人物 A の登場段落数}) + (\text{人物 B の登場段落数})}^{(1)}$$

「電気看板の神経」という作品を対象に、友好敵対関係を推定した結果、提案手法により正しく推定できたのは、13 箇所中 8 箇所 (61.5%) であった。成功例として「圭さんと吉公とはまあ仲のいい方で」という表現があり、失敗例として「鈴江は春江を殺しただけではなく、春江の情人たる岡安を完全に手に入れ」という表現があった。

6 考察

6.1 存在状態の判定について

死亡判定において、エラーの理由としては、婉曲な表現を判定できなかったことや、係り受け解析の誤りにより、適切な係り元と係り先の語句が抽出できなかったことが考えられる。死亡したことを正しく判定できなかった箇所の例として、「またまた惨劇が演ぜられた。不幸な籤を引きあてたのはふみ子という例の年増女給だった」という表現があった。逆に、生存者を死亡していると判定してしまった箇所の例として「木沢さん、死体を余りいじらぬように、傷を調べて下さい」という表現があった。ここでは、「木沢」という人物の係り先は「調べて」であると考えられるが、係り先を「死体」と解析していたのが原因であった。このような婉曲な表現や係り受け解析の誤りによるエラーは、パターンマッチングによる判定で精度を向上できると考えられる。

6.2 友好敵対関係の推定について

友好敵対関係の推定では、不正解のパターンは、一文に三人以上の人物が共起する場合で、関係者ではない人物間の関連度が、正しい関係者の関連度を上回ってしまっている事がエラー判定の原因となっていたため、Dice 係数以外の他の関連度算出方法を考慮することで、精度を向上できると考えられる。

6.3 存在状態の可視化について

今回の可視化では、スライダバーを動かすことで、各ページの情報を素早く読返すことができる。スライダバーの位置に応じて、リアルタイムに人物情報の可視化が行われるので、実際の利用シーンにおいて十分に有益な効果が期待できる。また、人物の存在状態という物語の柱となる部分を可視化により把握することで、今まで物語を把握できず、読書に対し億劫になっていた人に対する支援につながることや、文字ばかりのものに画像が加わることでより親しみやすくなる効果が期待できる。そこで、本手法による存在状態の可視化によって、実際に理解の支援になるかどうかを確認するための評価実験を実施する必要があると考えている。また、シルエット画

像はランダムに提示しているが、ユーザに与える人物の印象に影響すると考えられるため、人物の特徴等の抽出結果に応じた適切なシルエット画像の表示が必要であると考えている。推理小説の場合では、死亡したと思っていた人物が後で、実は生きていたという場合もあるため、一度死亡したと判定しても、生き返った情報を判定し可視化するということも必要であると考えられる。また、著者の意図的な誤誘導には、見破れるシステムを作るのではなく、ユーザの視点に合わせた可視化を実装することで、物語の楽しみを損なわずに読書できると考えている。

7 まとめ

本稿では、物語テキストにおける登場人物の存在状態を判定する手法と人物間の友好敵対関係の推定手法を提案し、人物の存在状態を物語の読む進み具合に応じて可視化した。今後は、友好敵対関係以外の人物関係も推定し、人物相関図を物語の進行に沿って、可視化することを目指す予定である。

参考文献

- [1] B. Coyne et al. WordsEye: a text-to-scene conversion system. In *Proc. of Conference on Computer Graphics and Interactive Techniques*, pp. 487–496, 2001.
- [2] 佐藤圭太, 西原陽子, 砂山渡. ブログ記事からのイベント文抽出によるシーンの生成. 第 21 回人工知能学会全国大会, 2007.
- [3] O. Akerberg et al. CarSim: An Automatic 3D Text-to-Scene Conversion System Applied to Road Accident Reports. *Proc. of Conference on Europe Chapter of the Association for Computational Linguistics*, vol.2, pp. 191–194, 2003.
- [4] K. Sumi et al. Automatic conversion from E-content into animated storytelling. In *Proc. of International Conference on Entertainment Computing*, pp. 24–35, 2005.
- [5] 松本 圭祐, 伊藤 雄一, 村上 礼繁, 北村 喜文, 岸野 文郎. テキストの内容理解促進を補助するための動的なマルチメディアコンテンツ生成に関する検討. 情報処理学会研究報告, vol.2006, no.24, pp. 121–128, 2006.
- [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ー関係図自動構築. 言語処理学会年次大会発表論文集, vol.14, pp. 380–383, 2008.
- [7] 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. 言語処理学会年次大会発表論文集, vol.13, pp. 574–577, 2007.
- [8] 芳村 知則, 延澤 志保, 太原 育夫. 共起パターンに基づく小説内登場人物の相関図作成. 電子情報通信学会大会講演論文集, vol.2007, p. 54, 2007.
- [9] 青空文庫. <http://www.aozora.gr.jp/>.
- [10] 松尾豊, 友部博教, 橋田浩一, 中島秀之, 石塚満. Web 上の情報からの人間関係ネットワークの抽出. 人工知能学会論文誌, vol.20, pp. 46–56, 2005.
- [11] あの一と検索スパイシー. <http://spysee.jp/>.

未来ビジョン

今回は、電子書籍の特徴を活かした理解支援の一環として、物語テキストのみを対象とし、物語の流れに沿って、登場人物の相関図を生成することで、物語の内容の理解を支援する手法について述べた。しかし、将来的には、様々な分野の人物関係の可視化にも応用できると考えられる。

例えば、ニュースを対象とし、複雑な政界の人物関係を時系列に沿って可視化したり、一人の人に注目し、時系列順にその人を中心に变化していく人物関係を可視化したりすることも可能であると考えられる。それにより、膨大な過去のニュースを確認して、自分の頭の中で整理するという手間の必要がなくなり、簡単に今までの流れを把握することができる。また、関係のみにとどまらず、評論本や専門書等の理解することが難解な内容について、論理の流れを可視化するといったことができると、より読書の支援につながると考えられる。

他にも、教育の業界にも発展可能であると考えている。例えば、国語の学習の際に、学習レベルに応じた適切な挿絵を挿入するという

ようなシステムが考えられる。読み進めるにつれて変化することが重要で、学習者のレベルに応じた変化や、難解な部分のみへの提示等により、教育の幅を広げられる。

また、画像や音声といったマルチメディアコンテンツを付加することにより、内容の理解支援を行うことで、読書が苦手な、今まで文字だけで読む気がおこらないといった理由や、状況を把握しづらいといった理由から読書から遠ざかっていた人達にとって、読書に親しむきっかけとなり、気軽に読書できるようになり、活字離れ等の問題も解決していけると考えられる。また、もともと読書が苦手だったという人でない場合でも、今までは自分にとって少し難易度が高く、なかなか手をつけられていなかった本を読むきっかけにもつながれると考えられる。

このように、今までにない様々な理解支援につながる機能を提供できると考えられる。電子書籍の可能性を広げ、紙の本では実現不可能であった、電子書籍の特性を利用することで、電子書籍や読書における今までの常識を新しく塗りかえられる研究に挑戦していきたい。