

物語の要約を支援するインタフェース

Interactive Narrative Summarization

田中 翔太郎 岡部 誠 尾内 理紀夫*

Summary. 本研究は、ユーザが物語の要約を作成する作業を支援するインタフェースを提案する。物語の要約は、読む前に内容を知りたいときや1度読んだ物語の内容を思い出したいときに需要があり、効率よく作成できると便利である。テキストの自動要約研究は多数存在するが、そのほとんどは論説文やニュース記事といったフォーマットの整ったものを対象としている。しかし、物語は作品ごとにフォーマットや文体が異なり、比喩表現や言い換えを多数含むため、既存手法の適用が難しい。そこで、システムが全自動で要約するのではなく要約作成を支援することで、物語を自動要約するときの問題を解消する。ユーザは、システムが表示したキーワードの中から物語の内容を反映しているものを選択する。そして、そのキーワードを含む文の中から要約に使用可能なものを選択していくことで、本文のリストを生成し、それをもとに要約を作成する。本システムは、情報検索の分野で広く利用されている tf-idf を利用して物語の内容を特に表現しているキーワードの抽出を行う。次に、キーワードからは読み取れない登場人物の関係といった情報を得るために、Simpson 係数を利用して抽出したキーワードと結びつきの強い語を調べる。ユーザスタディの結果、時間を一定にして要約の作成を依頼したところ、本を読む場合と比較して提案手法を用いた方が、良い要約を作成できた。

1 はじめに

世の中には様々な物語がある。この中から自分の読みたい物語を探す方法としては、Web 上のショッピングサイトのレビュー文や評価値を参考にするとともに、物語のタイトルやキーワードを検索にかけたり、出版社の記述したあらすじを読んだりして物語の内容を調べる方法がある。例えば、紀伊国屋書店の書籍検索 [1] では物語のタイトルだけでなく、目次やあらすじまで調べることができる。ただし、あらすじが存在しない場合や、出版社の記述したあらすじに自分の知りたい情報が含まれていない場合は利用できない。ここで、物語の要約を効率よく作成できると役立つ。過去に読んだ物語の内容を思い出すことや、物語の内容を高速に理解することも容易になる。

従来、テキストを自動的に要約する研究は多数存在する [2]。論説文やニュース記事を要約する手法としては、文書のタイトルに重みをつける [3]、文章の位置情報や手がかり表現を利用して冗長な語を削減する [4]、クラスタリングにより重要な語や文を抽出する [5] といった方法がある。しかし、これらの手法はそのまま物語には適用できない。物語は比喩表現や言い換えが非常に多く、重要な単語が何度も出てくるとは限らない。また、論説文やニュース記事と違ってタイトルが内容の要約になっていると

は限らない上に、フォーマットや文体も物語によってさまざまである。さらに、短編でも 1,000 文程度の文章量があるため、上記のような手法の適用範囲を超えている。その上、要約を作成する際に「玄関ドアも、ほかの部屋の窓も鍵がかかっていた」という表現を「密室」と言い換えるなど、作中には存在しない単語を用いることもある。以上のような理由から、既存の要約手法を適用して物語の要約を全自動で作成するのは難しい。

そこで我々は、自然な要約を作成するために、人間が正しい要約を作成する作業を支援するインタフェースを提案する。ユーザは、システムが表示したキーワードに注目する。そして、そのキーワードを含む文から要約で使用可能な文を選択していくことで、本文のリストを生成し、それをもとに要約を作成する。このようなインタフェースは我々の知る限りでは今までに存在しなかった。

本システムは、tf-idf を用いてキーワードの抽出を行い、Simpson 係数を利用して関連語を調べる。そして、ユーザが選んだ文を物語で登場する順にソートして表示する。本システムを利用することにより、より自然な要約を短時間で作成できると考えられる。実際、ユーザスタディによって、同じ時間だけ物語を読む場合と比較してより良い要約を作成できた。

2 関連研究

レビューサイトの評価値やレビュー文といった評価情報の分析に関する研究は多数存在し、大知らのレビュー文内の単語を新たな特徴量として採用して

Copyright is held by the author(s).

* Shotaro Tanaka, 電気通信大学 電気通信学部 情報工学科, Makoto Okabe, 電気通信大学 情報理工学部 総合情報学科・科学技術振興機構 さきがけ, Rikio Onai, 電気通信大学 情報理工学部 総合情報学科

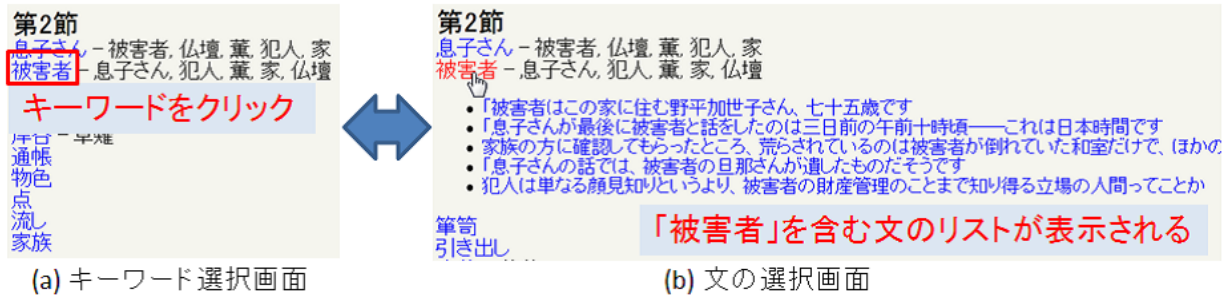


図 1. システムの出力画面

予測値の精度を向上させる研究 [6] や WordNet, 統計分析, および映画知識を用いた映画レビューのマイニングと要約を行う研究 [7] が挙げられる. これらの研究は, ユーザの意見や感想を知りたいときには有用であるが, 今回のようにコンテンツの内容を知りたいときには使用できない.

物語に関連した研究としては, 物語のデータベースをもとに簡単な童話を作成する研究 [8] がある. これは, あくまで新しい物語を作ることが目的なので要約には使用できない. また, 要約そのものをしていないわけではないが, 「お金持ちの商人」を「あの男」のように短く言い換える手法 [9] がある.

物語の要約の研究としては, 本の自動要約の研究 [10] がある. これは今まで使われてきた要約技術を改良して物語に応用しているが, 作成される要約の文章量が大きい. 我々は1つの物語をより簡潔にまとめたいため, この研究は使えない. また, 日本語の物語要約の研究として, 談話理解モデルに基づいた重要文抽出による要約 [11] 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, この数十倍から数百倍の文からなる長い物語の要約には使えない. また, 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 [12] は, 単語のリストの提示によりユーザが1度読んだ物語の内容を思い出せることを確認した. 本研究ではこの結果を踏まえ, キーワードのリストを提示し, ユーザに物語の内容を反映していると思われるものに注目させる. そして, そのキーワードを含む文を選択していくことにより, 読んだことのない物語でも要約を作成可能であることを確認した.

3 システム概要

ここでは, 我々のシステムを用いて物語の要約を作成する際の流れを述べる. まず, ユーザが物語のテキストデータを入力すると, システムは図1(a)のような画面を出力する. 各行の先頭にシステムが抽出したキーワード, その後ろにキーワードと結びつきの強い関連語が表示されている. 図1(a)中の「息子さん - 被害者, 仏壇, 薫, 犯人, 家」は, 「息子さ

ん」がキーワード, 「被害者」「仏壇」「薫」「犯人」「家」が「息子さん」の関連語であることを示す. ここで, ユーザは物語の内容を反映していると思われるキーワードに注目する. 例えば「被害者」は推理小説以外ではほとんど登場しない単語である. 関連語として「犯人」が含まれることから, 「被害者」は物語の内容を反映した重要な単語である可能性が高い. また, 物語の要約を作成する際には「誰が何をした」ということを正確に把握しなければならないため, 「堀部」といった人物名にも注目する. 注目するキーワードをクリックすると, そのキーワードを含む文のリストが表示される. 図1(b)は図1(a)において「被害者」をクリックしたときの画面である. この中から, ユーザは物語の要約として利用できそうな文を選択する. 例えば「被害者はこの家に住む野平加世子さん, 七十五歳です」を読むと, 被害者の名前を知ることができる. このように, 素早く特徴的な単語を見つけ, 特徴的な文にアクセスすることで, 物語の理解を助けることができる.

最後に, ユーザが選んだ文は物語で登場する順にソートされて, キーワードや関連語と共に表示される(図2). 例えば, 図2の1行目は「被害者はこの家に住む野平加世子さん, 七十五歳です」が選んだ文, 「被害者」が選んだキーワード, 「息子さん」「犯人」「薫」「家」「仏壇」が「被害者」の関連語である. ユーザはこの結果を上から順番に読んでいき, 物語の要約の作成を行う.

4 使用したアルゴリズム

4.1 名詞抽出

人や場所, 物事, 出来事などの名称を表す名詞は, 物語の特徴を反映した語である可能性が高いと考えられる. そのため, 今回は MeCab [13] を利用して名詞の抽出を行うことにする. この際「洗濯機」や「三億円」などの名詞は, それぞれ「洗濯」「機」, 「三」「億」「円」と複数の単語に分解されてしまうので, 今回は連続する名詞は1つの単語として扱うことにする. また「の」「こと」「もの」といった単体では全く意味をもたない単語は, 物語の要約に役

結果

選んだ文 - キーワード [関連語1, 関連語2, 関連語3, ...]

出典: 東野圭吾「ガリレオの苦悩」

「被害者はこの家に住む野平加世子さん、七十五歳です - 被害者[息子さん, 犯人, 薫, 家, 仏壇]
 貴美子は、犯行があったと思われる日に野平家を訪れたことを認めた - 貴美子[女性, 真瀬 草薙, 部屋, 薫, 後]
 真瀬葉月がダウジングを使って犬の死骸を発見したのは事実なんです - 死骸[犬, 発見, ダウジング, 彼女]
 碓井の部屋の天井裏から金の地金が見つかったため、自供するまでに時間はかからなかった - 碓井[時, 野平家, !]
 犯人には犬に噛まれた傷があるはず——お見事でした - 傷[犬, 碓井, 犯人, 死骸, 時, 場所, 彼女]
 鑑識で犬の死骸を改めて調べてもらおうと、歯から人間の血液が検出された - 歯

図 2. ユーザが選んだ文のリスト

表 1. 抽出したキーワード

1 節		2 節		6 節		7 節		8 節	
word	tf-idf	word	tf-idf	word	tf-idf	word	tf-idf	word	tf-idf
堀部	0.0337	息子さん	0.0140	湯川	0.0163	湯川	0.0158	碓井	0.0267
堀部浩介	0.0150	被害者	0.0117	ダウジング	0.0109	大学	0.0141	傷	0.0200
水品	0.0112	筆筭	0.0105	死骸	0.0063	婆ちゃん	0.0106	野平家	0.0089
堀部さん	0.0112	引き出し	0.0105	空間	0.0058	葉月	0.0088	湯川	0.0079
動き	0.0112	岸谷	0.0093	空中	0.0058	水	0.0071	死骸	0.0079
葉月	0.0110	通帳	0.0070	毒薬	0.0058	答え	0.0067	際	0.0067
胸	0.0075	物色	0.0070	発見	0.0054	振り子	0.0061	良心	0.0067
真瀬	0.0075	点	0.0070	結果	0.0051	薫	0.0054	科学	0.0067
深呼吸	0.0075	流し	0.0070	実験	0.0051	パイプ	0.0053	猫	0.0067
指先	0.0075	家族	0.0070	女の子	0.0051	死骸	0.0050	歯	0.0067

出典: 東野圭吾「ガリレオの苦悩」[17]

立たないので、不要語としてあらかじめ除去する。

4.2 tf-idfによるキーワード抽出

抽出した名詞の中から、キーワードの抽出を行う。ここで、物語中に何度も登場する語は登場人物の名前や舞台となっている場所であることが多い。また、物語中のある一部の場面でしか登場しない語は、その場面を特徴づける単語である可能性がある。そこで、本システムでは語の出現頻度が高いほど、登場する文書数が少ないほど値が大きくなる tf-idf を利用してキーワードを抽出する。

tf-idf は、語の出現確率と語の持つ情報量を掛け合わせた特徴量である [14]。情報検索の分野で広く使われおり、実際に Web からの人物に関するキーワード抽出にも使われている [15]。本システムは、入力として要約を作成する物語を与える。今回はあらかじめ複数の節に分かれている物語を使用し、1つの節を1つの文書と考えて tf-idf の計算を行う。一般に、tf-idf 値が高いほどその節で特徴的な単語である。今回は tf-idf 値の高い順に各節で 10 語ずつ抽出する。ここで、抽出した結果の例を表 1 に示す。なお、複数の節に分かれていない物語についても、10 ページおきなど機械的に複数の節に分割することでキーワードの抽出は可能である。

4.3 Simpson 係数による関連語抽出

tf-idf によるキーワード抽出を行うと、すべての節で登場する単語はキーワードとして出現しなくなる。そのため、「顔」「目」「今」などの一般的な単語は出現にくい反面、主人公の名前や何度も登場する物語のキーワードまで出現しなくなる恐れがある。そのため、本システムは tf-idf で抽出したキーワードと結びつきの強い関連語を同時に表示する。例えば、tf-idf 値の高い「湯川」の関連語は「後」「薫」「発見」である(表 2)。ここで、tf-idf 値は「後」が 0.0011、「薫」が 0.0037、「発見」が 0.0054 で、「後」と「薫」については、tf-idf 値の上位 10 単語に入らないため、キーワードには出現しない。しかし、「薫」は「湯川」と共に使われる人名で、作中も共に行動していることが多い。つまり、関連語を見ることで登場人物の関係を知ることができる。

単語 X と単語 Y の関連性の強さは、共起関係により調べる。共起関係とは、単語 X がある文書中に出たとき、その文書中に単語 Y がどの程度出現するかというものである。ここで、単語 X が登場する文書数を $|X|$ 、単語 Y が登場する文書数を $|Y|$ 、単語 X と単語 Y が同時に現れる文書数を $|X \cap Y|$ とする。今回扱う文書は 1 つなので、行頭が空白の文から末尾に改行を含む文までの複数の文のまとまり、

表 2. キーワードとその関連語

キーワード	関連語
湯川	後, 薫, 発見
ダウジング	結果, 実験, 話, 彼女, 死骸, 犬, 発見
死骸	犬, 発見, ダウジング, 彼女
発見	犬, 死骸, 女の子, 湯川, ダウジング
結果	ダウジング, 実験, 話
実験	ダウジング, 結果
女の子	発見

出典：東野圭吾「ガリレオの苦悩」[17]

つまり形式段落を1つの文書として扱い、同じ形式段落で登場する単語同士の結びつきを考えることにする。

共起を調べる係数としては、相互情報量、ダイス係数、Jaccard 係数、Simpson 係数などが挙げられる。物語では、単語ごとに出現回数が大きく異なる。例えば、人名の出現回数は他の単語と比較すると多くなりやすい。そのため、 $|X|$ と $|Y|$ の差が大きいと、関連性が高くても係数が小さくなる欠点をもつ相互情報量、ダイス係数、Jaccard 係数は利用できない[16]。よって、本システムでは式(1)で計算される Simpson 係数を利用する。

$$Simpson(X, Y) = \frac{|X \cap Y|}{\min(|X|, |Y|)} \quad (1)$$

一般に、Simpson 係数が高ければ高いほど、両方の単語の関連性が強いことを示す。ただし、 $|X|$ または $|Y|$ の値が非常に小さいと、関連性がさほど高くなくても Simpson 係数が大きくなる欠点があるので、適切な閾値を設定しなければならない。今回は単語 X, 単語 Y とともに 4 回以上登場しているときにのみ、Simpson 係数を計算を行うことにする。

5 実験

本システムの有用性を確認するために、本システムで作成した要約と実際に物語を読んで作成した要約の比較を行う。本システムは効率よく要約を作成することが目的なので、システムを何十分も使用しては意味がない。そこで、システムを使用する時間と実際に物語を読む時間に一定の制限を設ける。また、本システムは要約で使用する文の選択のみを行うため、今回の実験では選択した文のリストをもとに要約を作成する時間については考慮しない。要約の比較は、ユーザによって言い回しや文末表現、着眼点が異なるために簡単に行うことができない。そこで、本システムを使用したことのない被験者に要約の順位付けを依頼し、順位の高いものほど良い要約とした。

5.1 要約の作成

最初に、被験者を2つのグループに分けて実際に要約の作成を行う。今回の実験には、東野圭吾「ガリレオの苦悩」[17]を用いた。これは全部で5つの短編からなる推理小説で、今回は第4章の「指標す」の要約を行う。なお、第4章は全部で43ページで8つの節に分かれている。ここで、グループ1は本システムを利用して要約を作成し、グループ2は実際に本を読んで要約を作成する。今回は、どちらのグループも5分間で要約に使用する文の選択を行い、それをもとに要約の作成を行う。この実験には、各グループ5人計10人の20代学生が参加した。彼らは本システムを使用したことはなく、要約する物語を読んだこともない。また、要約を作成するときは、選択した文の内容の同一性を保つように依頼した。つまり、被験者は内容の変わらない言い換えと文の並び替えのみを行い、被験者の推測や感想、煽り文句を書かないこととする。具体的な実験手順としては以下の通り。

グループ1

1. 別の物語データを使って、システムを使う練習を行う(要約は作成しない)。
2. システムを5分間だけ使用して、要約に必要な文を選択する。
3. 選択した文のリストを用いて要約を作成する。

グループ2

1. 本を5分間だけ読んで、必要な文をコピーする。
2. コピーした文を用いて要約を作成する。

ここで、実験で得られた要約の一部を表3に示す。

5.2 要約の比較

本システムによる要約作成が有効であるかを検証するために、作成した要約の評価を行う。今回は、要約を作成した人とは別の10人に実際に物語を読んでもらい、要約に順位をつけてもらった。また、要約の評価基準は以下の通り。

- 簡潔にまとまっているか
- 物語の内容をよく把握できたか
- 物語の内容と異なっている点はないか

なお、要約の内容の比較をするために、文章の上手さおよび誤字脱字は採点基準には含まないこととした。このとき結果を図3に示す。横軸がユーザ名、縦軸が要約の順位、誤差棒は標準偏差である。ここで、ユーザ a, b, c, d, e はシステムを利用して要

表 3. 実験で得られた要約

ユーザ名	要約
a (システム)	強盗殺人が起こる。被害者は野平加世子 (75) で、筆筒から通帳と宝石、貴金属類が盗まれる。解決の手がかりとなったのは、ダウジングで発見された洗濯機の中に捨てられていた犬の遺体であった。犬の死骸を調べると歯から人間の血液が検出され、それが犯人の行動とつながったからである。容疑者の部屋の天井裏から金の地金が見つかったため、証拠は動かぬものとなった。
h (実際に読む)	彼女は堀部浩介に電話で呼び出された。用事の内容は察しがついていた。堀部が指定したのはファーストフード店で彼女の後に堀部が現れた。彼は長身で痩せているが姿勢はあまりよくないがサッカーをするときだけは力強い動きをする、そんな彼に彼女は魅かれていた。用件は彼からの交際の申し込みであったが彼女は一時保留した。母と二人暮らしのアパートに戻り、考えたうえで彼に断りの電話を入れた。

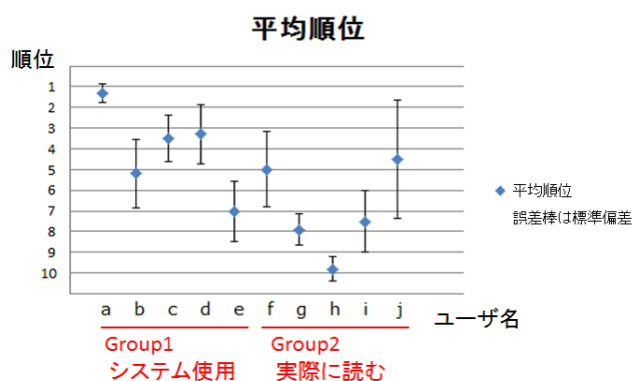


図 3. ユーザスタディの結果

約を作成し、ユーザ f, g, h, i, j は実際に本を読んで要約を作成した。

1 番良い要約は、システムを利用して作成したユーザ a の要約で、10 人中 7 人が 1 位をつけた。発生した事件の概要、解決の手がかり、証拠といった物語のポイントをしっかりおさえ、簡潔かつ正確にまとめられている。被験者からも起承転結がしっかりしている、簡潔でしかも重要な点をもらさず要約しているといったコメントが得られ、システムを使って重要な文を抽出し、要約を作成できたことが分かる。

1 番悪い要約は、本を読んで作成したユーザ h の要約で 10 人中 9 人が 10 位をつけた。肝心の事件に全く触れていない、物語の内容が途中だからといったコメントが得られた。実際、この要約を作成したユーザ h は 5 分間で最初の 1 節しか読むことができず、要約にもそれ以降の重要な情報が一切入っていない。

それぞれのグループ全体での平均順位は、システムを使って作成したグループ 1 が 4.06 位、本を読んで作成したグループ 2 が 6.94 位となった。被験者 10 人の順位データをグループごとに並べて Mann-Whitney の U 検定を行った結果、提案手法と本を読んだときの効率の違いが有意に示された。以上の

結果から、我々の手法は、同じ時間だけ本を読んだときと比較して良い結果を得られたと言える。

6 まとめと展望

本研究では、システムが提示したキーワードを含む文を選択していくことで文のリストを生成し、それをもとに物語の要約を作成するインタフェースを提案した。これにより、効率よく物語の要約を作成できるようになった。

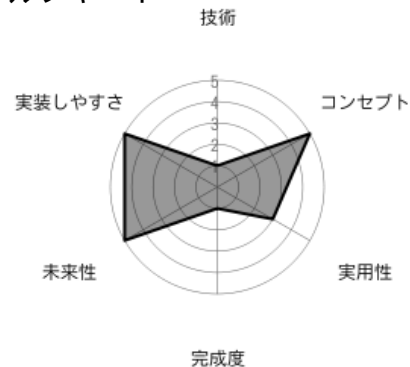
一方で、現在の実装では登場人物同士の関係が見えにくいという問題点が指摘された。これは、「彼」や「お前」などの代名詞を固有名詞に置き換えたり、別途、登場人物のリストを提示したりすることで、情報不足を補うことができる。また、キーワードと関連語以外の情報がほしいという要望もあった。例えば、システムに「葉月」と提示されても、それが人物名か場所名か 8 月なのかが分からないと、「葉月」を含む文のリストを読んで、そのキーワードがどのように使われているのかを確認しなければならない。もし、それぞれのキーワードの使われ方が最初から分かれば、要約に使う文の選択にかかる手間を軽減できる。今後はインタフェース部に改良を行い、様々な物語に対して比較実験をしていきたい。

参考文献

- [1] 紀伊國屋書店 BookWeb 和書詳細検索. <http://bookweb.kinokuniya.co.jp/guest/cgi-bin/wshosc.wb>
- [2] Knight, K. and Marcu, D. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. In *Artificial Intelligence*, vol.139, no.1, pp.91-107, 2002.
- [3] 大森岳史, 増田英孝, 中川裕志. Web 新聞記事の要約とその携帯端末向け記事による評価. 情報処理学会自然言語処理研究会, NL-153, pp. 1-8, 2003.
- [4] 船坂貴浩, 山本和英, 増山繁. 冗長度削減による関連新聞記事の要約. 情報処理学会研究報告. 自然言語処理研究会報告, vol.96, no.65, pp. 39-46, 1996.

- [5] Zha, H. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.113–120, 2002.
- [6] Ochi, M. and Okabe, M. and Onai, R. Rating prediction using feature words extracted from customer reviews. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, pp.1205–1206, 2011.
- [7] Zhuang, L. and Jing, F. and Zhu, X.Y. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pp.43–50, 2006.
- [8] McIntyre, N. and Lapata, M. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp.1562–1572, 2010.
- [9] Hervás, R. and Finlayson, M.A. The prevalence of descriptive referring expressions in news and narrative. In *Proceedings of the ACL 2010 Conference Short Papers*, pp.49–54, 2010.
- [10] Mihalcea, R. and Ceylan, H. Explorations in automatic book summarization. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp.380–389, 2007.
- [11] 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.
- [12] 藤井崇介, 土井昇一郎, 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.
- [13] MeCab. <http://mecab.sourceforge.net/>.
- [14] 相澤彰子. 語と文書の共起に基づく特徴度の数量的表現について. 情報処理学会論文誌, vol.41, no.12, pp.3332–3343, 2000.
- [15] 森純一郎, 松尾豊, 石塚満. Web からの人物に関するキーワード抽出. 人工知能学会論文誌, vol.20, no.5, pp.337–345, 2005.
- [16] 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, no.1, pp.46–56, 2005.
- [17] 東野圭吾. ガリレオの苦悩. 文藝春秋, pp.199–241, 2008.
- [18] Ikeda, K. and Hijikata, Y. and Nishida, S. Proposal of Deleting Plots from the Reviews to the Items with Stories. In *Database Systems for Advanced Applications*, pp.346–352, 2010.

アピールチャート



未来ビジョン

物語を読む前に内容を知っても気にしない人もいれば、先に知ってしまう、いわゆるネタバレを気にする人もいる。現在の実装では、簡単に結末が分かってしまうため、ネタバレを気にする人は読む前の内容確認として使うことはできない。そこで、ネタバレを避けるために途中までの要約を作成するようにしたり、レビュー文からあらすじを自動的に除去し、意見部分を抽出する研究 [18] のように、物語の核心をつく部分を隠したりする方法が考えられる。

本システムは、高度な技術を使うことなく実現可能なため、技術的なすごさはない。一方で、プログラミングをしたことがある人ならば誰にでも作れるほど実装しやすく、扱いやすい。例えば、今回はキーワードとして各節で 10 単語ずつ出力したが、5 単語ずつや 100 単語ずつに変更することも容易である。また、テ

キストデータさえあれば、どんな物語もすぐに解析できる。

本システムを使うと、物語を読んでいなくても効率よく内容を理解できるので、「どうせ読むのであれば面白い本を読みたい」という簡単に叶えることができる。しかし、ネタバレを気にする人や本を全く読まない人は使えないため、その点では実用性が下がる。そこで、今後はネタバレ防止機能をつけるなど誰もが使える、使いたくなるシステムを目指して改良を重ねる予定である。また、現在の機能に加えて、登場人物同士の関係を出したり、代名詞を置き換えたり、選んだ文から要約を作成することを支援したりできるようにして、システムの完成度を高めていく。将来的には、テキスト処理とインタフェースという今までにあまりなかった組み合わせを生かし、ユーザとシステムが対話をすることで、誰もが自分の目的あった要約を簡単に作成できるようにしたい。