物語の時系列要約法

呉宗翰　天沼博　松澤和光
神奈川大学大学院　工学研究科
r202170128yf@jindai.jp amanuma@kanagawa-u.ac.jp　matsuk90@jindai.jp

概要

本研究では、アニメで放送されたライトノベル作品に対して、時系列を考慮した小説自動要約を提案する。第1章では、コミックやアニメ、ライトノベルの問題点について述べる。第2章では、自動小説要約における既存研究について簡略的に述べる。第3章では、本研究で提案するシステムについて述べる。第4章、第5章では、第3章で提案したシステムについて評価、考察を行う。

1 研究目的

多くのライトノベルは、制作会社によってアニメ化、コミック化されている。そのため、原作を知らなくとも制作された分だけライトノベルの物語を知ることができる。しかし、放送を待てずに物語の続きを知りたい人がライトノベルを購入しようにも、続編が何巻に相当するのか、分かりづらい。また、アニメ制作会社が続編を放送するのを期待しても長い月日がかかる。さらにコミックでは、打ち切りといったことも起き、続編が描かれない場合もある。

また、ライトノベルは基本的には時系列順に物語を進めていくが、作者が意図して時系列を順序不同にすることがある。『空の境界』の作者、奈須きのこ氏によると、時系列を順序不同にすることで、物語がミステリ的な内容になり、読者は物語を考察・整理する事で快楽を得られる[1]。しかし、パズルのような作品を読むのが苦手な人にとっては、読みづらく購入しづらい。そこで、ライトノベルを楽しみ易くするために、物語を時系列に並び直した情報を提示して物語を読む助けとする。さらに、物語の重要な事柄を抽出する事で、アニメや漫画の続きを読むためにライトノベルを購入する人の手助けになると考える。

2 既存研究

自動要約モデルは新聞や論文向けのものが多い[2][3][4]。小説の内容に焦点にあてて研究しているものの多くは、「走れメロス」や「羅生門」といった純文学が対象となっている。ライトノベルや推理小説等といった娯楽小説（大衆文学）の自動要約については、ごく少数の研究者が行っているだけである。

3 提案手法

システム全体のフローチャートは図1に示す。また、対象となる小説の物語文には、テキスト処理を行い「代名詞」を「固有名詞」に変換したものを使用する。

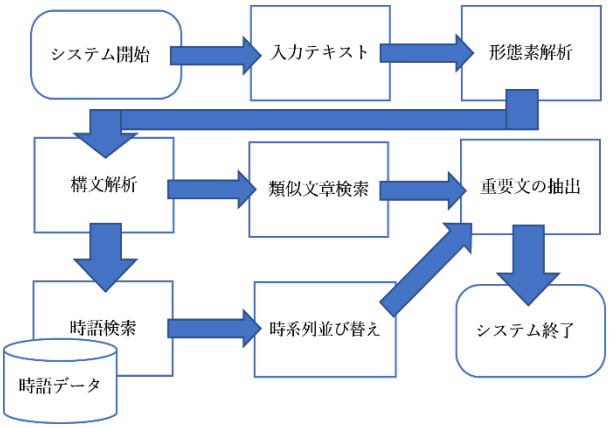

図 1　システムのフローチャート

3.1 品詞や単語の選別

ライトノベルには、情景や感情を表すために、「きれい」や「はぁ」といった形容詞や感嘆詞の等の品詞を多く使われる。しかし、ライトノベルの内容の大まかに知る上で不必要な品詞であるため、省くことでライトノベルの文章量を減らす。また、ライトノベルには架空の人物や国等が登場するが、Mecabのシステム辞書では、それらの単語を未知語と表記され、単語に正しい品詞が生成されない。そのため、ユーザー辞書を追加することで対応させる。

3.2 文の主節を抽出

小説には、登場人物の「感情」や「情景」を強く強調するために、「比喩」や「倒置法」「擬人法」といった修辞法を使われている。修辞法を使用した文は、結論がぼやけてしまい、内容を解りづらくしてしまう。特に、ライトノベルは「倒置法」「比喩」を使い、登場人物の魅力を上げているので、文章の内容をわかりやすくするために、文の主節を抜き出すのは必須だと考える。

今回は、山内長承氏が開発したシステム[5]を参考に、主語、動詞、目的語、情報句を抜き出す。情報句とは、「日時」や「場所」といった、文には影響しないが情報となる句のことである。この情報句を使い、3.5 節に示す時語検索を行う。

その他の主節は、主語、目的語、動詞の順に一つの文にする。その後、この文を使って類似文章検索、重要文抽出を行う。

3.3 類似文章を省く・要約する

機械学習ライブラリである「scikit-learn」を使い、文章に tf-idf 値をつける。ここで、scikit-learn で行う tf-idf の定義[6]は一般に知られている定義とは異なるので以下に示す。

$$tf_{t,d} = \text{文書 d 中の単語 t の出現回数}$$

$$\mathrm{idf_t} = \log\frac{1 + \text{総文書数}}{1 + \text{単語 t を含む文書数}} + 1$$

また、類似度はコサイン類似度で測定する。

ライトノベルでは、主に地の文と会話文で構成され、地の文と会話文で同じ内容を表現の仕方を変えて描かれていることが多い。そこで、似た内容の文章を以下のように省く。

まず 1 文目と 2 文目以降を比較し、1 文目と類似性が高いと評価された文を省く。次に 2 文目と 3 文目以降を比較し、2 文目と類似性が高いと評価された文を省く。同様に、3 文目以降もこの操作を行う。

3.4 重要文の抽出

重要度の高い文を抜き出すことで、アニメやコミックからその作品を知った人でもライトノベルを途中から読み始めることができる。ここで、重要度の高い文とは、主人公に降りかかる出来事や、主人公に間接的にかかわる出来事と仮定する。ライトノベルでは、主人公に関係する出来事には、主人公や主人公に近しい人物の名前が現れる。それらを抽出する事で重要度の高い文を抽出できると考える。

重要度を測るためにテキスト自動要約ライブラリである「pysummarization」を使用する。要約ライブラリの制作者 Accel Brain 社によると、このライブラリは「LSTM をベースとした Sequence-to-Sequence の学習を実現するニューラルネットワーク言語モデルの基礎モデルを re-seq2seq の学習モデルや再構成モデルに応用することで、文書自動要約器を実装したもの」[7]と説明されている。

このライブラリで算出した重要度のうち、主人公や主人公に近しい人物の名前がある文の重要度を高くする。その後、重要度の高い文、10～12 個を出力する。

3.5 時語検索

時語データを作成する。作成方法は先行研究[8]を参考にする。奥村紀之氏は時間に関する情報を保持した時語データベースとして，時間帯単位，年月日単位，季節単位，人生単位の 4 種類に分類した。以下に，奥村氏による 4 種の単位の定義と例を示す[8]。

- 時間帯単位時語
  一日の中のある時間帯を表す時語
  (例：朝，夜)
- 年月日単位時語
  一年や一日単位の移り変わりを表す時語
  (例：日，年，今日，再来月，火曜)
- 季節単位時語
  一年の季節のどこかを表す時語
  (例：春，夏，春分，七夕，海)
- 人生単位時語
  人の一生のある期間を表す語句
  (例：高校生，生年)

辞書作成にあたり奥村氏の方法は「時を関する語句をアンケート調査でサンプル収集」→「サンプルを 5 名の被験者が判定」→「時間関連の語句であると評価」したものを時語として採用する[8]。今回は時間の都合上、「辞書で時間に関する語句を収集」→「被験者 132 名で 15[%]以上が認めた時語」→「国語の高校教員 2 名に妥当と評価」したものを時語として採用したものを使用する。また、人生単位時語は評価しない。

3.6 時系列に並びなおす

3.2 節で抽出した情報句を日数と時語に分ける。時語は、「年月日」「時間帯」「季節」の 3 項目を判定させる。

まず初めに物語の始まりの文を「0 日」とおく。それ以降の文で「90 日」や「『年月日』に関する時語」があれば日数計算を行う。その後、日数の数が低い順に並べる。

その他の 2 項目では、「時間帯」は「朝」「昼」「夕」「夜」を、「季節」は「春」「夏」「秋」「冬」をそれぞれ順番に「1」「2」「3」「4」のラベルを付ける。その後、日数を基準に「時間帯」は 1 日ごとに、「季節」は 365 日ごとにラベルの数が低い順に並べる。

4 評価実験

今回提案した手法の結果を図 2 に示す。この結果をもとに 2 種類の評価実験を行う。

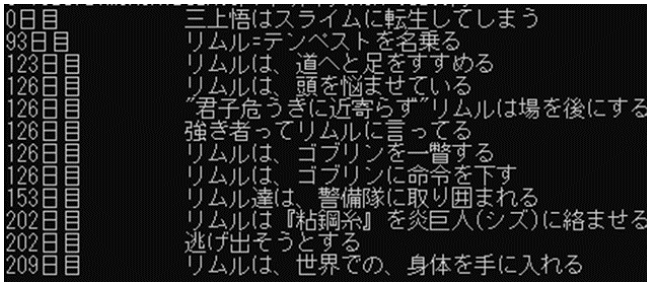

図 2 システムの出力結果

4.1 正誤評価

アニメ 1 話ごとを理想的な出力結果と仮定し、出力結果と比べ正誤評価を行う。評価項目は、記述の精度、あらましの再現率、話のバランス、の 3 項目を行う。記述の精度は出力結果の記述が適当かを判定、あらましの再現率はアニメの何話を再現しているかを判定、話のバランスはアニメ全話で偏りがないか判定する。正誤結果の一部を表 1 にまとめた。正誤結果は記述の精度は 21.7[%]、再現率は 52.2[%]である。

表 1 アニメとの正誤判定の一部

アニメの話	1	2	3	4	5	6
記述の精度	正	正	誤	誤	誤	誤
あらましの再現率	正	正	誤	正	誤	誤
話のバランス[個]	1	1	2	1	0	0

また、出力された日時が適正かライトノベルと比べた結果、アニメの一期の最後と比べ、6 日のずれが生じた。図 3 にシステムと原作小説の経過日数の比較グラフをしるす。

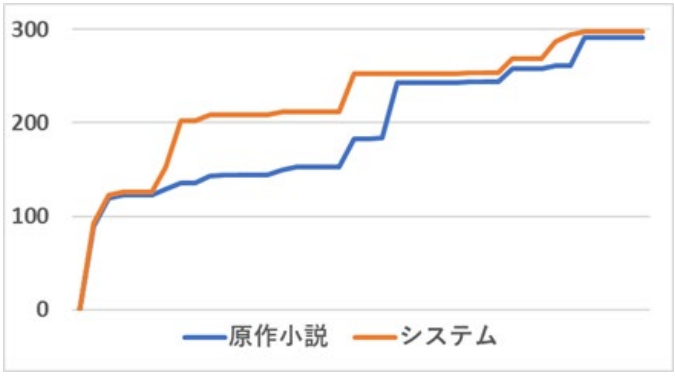

図 3 原作小説とシステムの経過日数

4.2 アンケート評価

アニメ視聴者にアンケートを行い、あらましを把握するのに役立つか、またアニメ何話なのか分かる出力になっているか評価してもらう。評価結果を図 4、図 5 にまとめた。

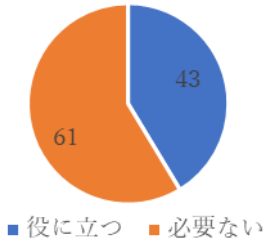

図 4 あらましの把握

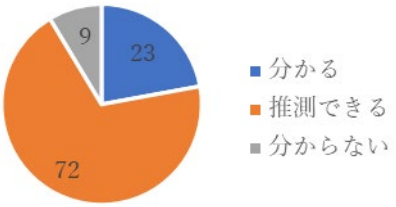

図 5 小説の話の推測

5 考察

今回、提案した手法により、アニメの続編にあたる話を原作小説から読むことが可能であることがわかる。このことは、アンケート評価から分析できる。

あらましの把握では、必要がないと回答した被験者が多かった。必要がないと回答した理由の多くが「意味の分からない文が多い」と否定的であった。一方で、「あらましの把握には役に立たないが、日

数と文からある程度、想像できる」と肯定的な理由があった。

小説の話の推測では、「推測できる」と回答した被験者が多く、その理由が「日数と内容から推測できるが、自信をもっては答えられない」であった。

このことから、アニメ放送後の内容の続きから原作小説を読むことは可能と考える。ただし、あらましの把握で否定的な理由が「不可解な文」であることから、この手法は、記述の精度が不十分である。このことは、正誤評価から考察される。

前述した通り、誤評価では特に記述の精度が悪い。その結果、あらましの再現率が低なり、話のバランスに偏りが出たと考えられる。その原因は二つ挙げられる。

1つ目は、「助詞」である。今回行った構文解析では、「述語」に対しての「助詞」の関係で、「主語」と「目的語」を判別していた[5]。しかし、「主語」や「目的語」につながる「助詞」の有無を判別しなかったことで、文章として不十分な文が出力された。

2つ目は「要約方法」である。アニメは、製作費等によるが、3ヶ月を1クールで全12話を1話25分で放送し終了する。そして、小説によって、全12話で放送される小説の内容が1章から2章で構成される。しかし、制作会社は1話25分を全12話で原作小説の1章から2章に描かれている内容のすべてをアニメに反映することができない。そのため、不必要な内容を省略する場合がある。このことを考慮せずに原作小説の要約を行うと、アニメに描かれていない場面を、重要度の高い文として要約してしまう可能性がある。これを解決するには、入力した文章をそのまま出力される「抽出型」ではなく、入力した文章を元に新しい文を生成する「生成型」であるべきと考える。

最後に時系列のずれを考察する。提案した手法は、小説の始まりの順に計算されている。しかし、原作小説は、同じ内容を「視点」や「場面」を変えて描く場合がある。

「視点」とは、話の中心人物である「主人公」に起きた出来事を主軸に描くが、主人公以外のキャラクターの魅力を引き出すために、別の人物の「視点」で「主人公」に起きた出来事を描かれる。そのため、「主人公」に起きた出来事と別の人物の「視点」の2つ以上の「視点」に同じ内容を描かれる。しかし、今回の提案手法では「視点」について考慮しなかったので、経過日数にずれが生じた。

「場面」とは、話の中心となっている「舞台」や「場所」のことである。仮に「A国」「B国」「C国」とあり、「B国」「C国」から「A国」に行くのにかかる経過日数が「B国」の方が「C国」と比べて2日早いと仮定する。このとき、今回提案した手法では、実際には違う「国」を同じ「国」と取り扱ったために、経過日数にずれが生じた。

6 今後の課題

5章より、記述の精度と日数の計算方法が提案した手法の問題点である。

記述の精度は4.1節の結果で示した通り、記述の精度が21.7[%]と低い値となっている。また、4.2節で被験者から頂いた否定的な意見には「不可解な文が多い」「読みづらい」「内容がわからない」等々、文の記述に関する内容が多くみられた。

日数の計算方法は4.1節の図3により、提案した手法のシステムには、原作小説と比べて日数の経過が大きくなっている箇所があり、また変化していない箇所がある。これらの箇所は5章で述べた通り、「視点」や「場面」が変わった箇所であった。そのため、視点の変化や場面の変化を考慮する必要がある。

以上のことから、記述の精度と日数の計算方法については今後の課題である。

参考文献

[1]「とらだよ。vol.81」,とらのあな, 2007 年 10 月
[2] 安武 凌, 野中 健一, 岩井 将行「LexRank を用いた小説文章からの自動要約手法の検討」
第 16 回 Web インテリジェンスとインタラクション研究会
https://www.jstage.jst.go.jp/article/wii/16/0/16_38/_pdf/-char/ja(参照 2022/1/5)
[3] 下窪 聖人「BERT で獲得する各場面の分散表現を用いたコサイン 類似度に基づく小説の挿絵推」
法政大学学術機関リポジトリ
gradcis_16_19T0010 (2).pdf(参照 2022/1/5)
[4] 山本 悠二, 増山 繁, 酒井 浩之「小説自動要約のための隣接文間の結束性判定手法」
言語処理学会年次大会発表論文集
https://www.anlp.jp/proceedings/annual_meeting/2006/pdf_dir/C5-4.pdf(参照 2022/1/5)
[5] 東邦大学理学部情報科学科山内のサイト「CaboCha による係り受け解析の利用～文の主節の骨組を取り出す」
https://pepper.is.sci.toho-u.ac.jp/(参照 2022/6/23)
[6] scikit-learn“6.2.3.4.Tf–idfterm weighting”
6.2. Feature extraction — scikit-learn 1.2.0 documentation (参照 2022/6/28)
[7] pysummarization・PyPI,
「 Usecase: Summarization with Neural Network Language Model.」
pysummarization · PyPI（参照 2022/8/29）
[8] 奥村 紀之, 瀧本 洋喜,「物語文章における時系列推定の拡張」
第 12 回情報科学技術フォーラム
F-010.pdf (ieice.org) (参照 2022/6/30)