

# IDE 的支援を用いた小説執筆のためのアプリケーション構築

西原 将平\* 三浦 元喜†

## 概要.

本稿では、アマチュアによる小説執筆を支援するための方法を提案する。近年では、アマチュアであっても、Web 上で気軽に自らの小説作品を公開できるようになった。作品を公開することにより、作者は読者からのフィードバックを受けることができる。しかし、小説を書くことは、設定の管理をはじめ様々なタスクを必要とするため、特にアマチュアにとっては良い小説を作るのは困難である。それらの負担を軽減するために、我々は、自動キーワード抽出を中心とした、総合開発環境 (IDE) 的支援を行う手法を提案する。自動キーワード抽出は、本文内における登場人物の名前を検索し、人物リストへの追加を行う。登場人物は普通、作中の設定を含んでいる。それを自動的にリスト化することにより、作者が行う作業の手間を減らし、簡単に小説の整合性をチェックすることができる。我々は、Web アプリケーションとしてこれらの機能を実装した。

## 1 はじめに

近年、インターネットの発展により、各種の創作活動について、誰でも手軽に自らの作品を Web 公開することが可能な環境がつくられてきた。小説の執筆も創作活動のひとつであり、小説投稿サイトには数多くのアマチュアによる作品が掲載されている。一方で、小説の執筆は、慣れないうちには困難な作業であり、良い作品をつくることは難しい。

作者は、本文を書く以外にも、それに付随する設定の管理や読み返しなどを行う必要がある。これらの作業を疎かにすると、設定の矛盾を始めとしたミスが発生する原因となってしまう。当然、作品が長期化するにつれて、これらの作業にはより時間がかかるようになり、煩雑になっていく。それに伴い、本文を書くための時間が減少してしまうことになる。

これらの手間を削減することができれば、作者は本文の執筆に集中することができるようになると考えた。そこで、我々は、プログラミング開発における eclipse のような、IDE 的支援を小説執筆に当てはめた手法を提案する。これにより、統合性の確認を効率的に行うことができるようになり、作者の負担が軽減されることが考えられる。また、我々は、これらの機能を提供するための、Web アプリケーションの提案を行った。このアプリケーションは、設定管理の手間を削減するために、コメント機能、キーワードのハイライトと、それをより生かすためのキーワードの自動抽出機能をもつ。

## 2 機能

以下、アプリケーションがもつ機能について説明する。これらは主に JQuery で実装され、エディタ部分は CodeMirror[2] を用いている。

コメント機能は、本文中にコメントを残すことを可能とする機能である。執筆の際、前回の執筆から時間が開いてしまった場合、書こうとしていた内容や、どのように展開させるつもりだったのかという情報を忘れてしまうことが起こりうる。本文中にコメントを挟むことで、そのような事態を減らす効果が考えられる。図1のように、コメント化には@マークを用い、コメントアウトされた文章は文字色が変更される。

図2は、ハイライト機能を用いた場合のエディタである。ハイライト機能は、IDE における変数のハイライトと同様、本文中の単語を色分け表示する。ハイライトしたい文字列を選択後、ユーザは背景色と文字色を選択できる。同時に複数の単語をハイライトすることも可能である。単語を効果的に色分けすることにより、読み返しの際に確認をやりやすくなることが考えられる。

また、我々はキーワードの自動抽出機能を実装した。小説におけるキーワードとは、その作品の設定を多く含んだ言葉であると仮定することができる。多くの場合、作中で用いられる固有名詞がそれにあたる。我々は今回、その中でも登場人物の名前をキーワードと定めた。

キーワードの抽出には Mecab[4] を用いた。本文に形態素解析を行い、その中から名詞を抽出する。通常、固有名詞は本文中で繰り返し用いられるため、登場回数が一定以上の名詞のみを抽出する。抽出された名詞は、図3で示されるキャラクタテーブルにまとめられ、執筆画面に表示される。このテーブル

Copyright is held by the author(s).

\* 九州工業大学

† 九州工業大学 基礎科学研究系

は、キャラクタの名前と基本情報を表示する。現時点では、アイテム作成時には名前のみが入力され、その他の情報はユーザが望むならば編集可能となっている。

テーブルに入力されたキーワードは、そこから直接ハイライト機能を利用することができる。各アイテムに自動的に追加されるサーチボタンをクリックすることにより、その名詞のハイライトが行われる。

男が眉をあげた。  
「そうですか」  
「そうさ、オタワルの貴人として敬われないとなると、君の目には、私はたふっと、男が顔をほころばせたので、ホッサルは驚いた。  
男はしばらく黙っていたが、やがて、口をひらいた。  
@@ここにコメントを挿入  
「私は、あなたをただの若造などとは思っておりません」  
ホッサルは顔をしかめた。  
「やめてくれ。そんな風に言われると、よけいに自分が若造に思えてくる」  
むっつりとした顔で応えてから、ホッサルは大きく息を吐いた。

図 1. コメント機能

メロス は激怒した。必ず、かの邪智鼻虐じゃちぼうぎなれた此このシラクスの市にやって来た。メロス には父なるばる市にやって来たのだ。先ず、その品々を買集め、歩いているうちにメロスは、まちの様子を怪しく思った。か、二年まえに此の市に来たときは、夜でも皆が歌をうた。老翁は、あたりをはばかる低声で、わずか答えた。

図 2. ハイライト機能

id	名前	年齢	性別	職業	プロフィール	検索	編集
0	メロス	未設定	未設定	未設定	未設定	Search	編集
1	老翁	未設定	未設定	未設定	未設定	Search	編集

図 3. キャラクターテーブル

### 3 予備実験

固有名詞の抽出精度を上げるため、予備実験を行った。名詞を除外する規定回数を、本文の文字数からの割合で三種類定め、異なる4つの作品にそれぞれ適用した。作品には、「走れメロス」の全文約10,000文字、「鹿の王」より二人の主要人物の会話約1,500文字、および投稿サイトに掲載されたアマチュアの作品から二つ（それぞれ一部シーンの抜粋、数名による会話約6,000文字と、大人数によるアクション約9,000文字）を用いた。結果、規定回数を大きく上げた場合であっても、重要人物はほぼ抜き出されることが分かった。一方、シーンにおける登場人物が多い場合には、除外の規定回数を少なくした場合でも上手く抜き出せない傾向があることも発覚した。これを解決するために、シーン単位の頻度のみではなく、作品全体での登場頻度やその人物の行動などによる重みづけを行うことで、これらの作品でも精度を上げることができるのではないかと考え

る。また、同様に、上手く名詞として抜き出せない人物名もいくつか存在した。例えば和名では、「西」「原」など、意味のとれる最少単語が分離して抜き出されてしまう。これを解決するためには、辞書データの補強及び、特定のパターンではひとまとめの単語として抜き出す処理を行う必要があると考える。

今後は、予備実験により得られたデータ及び自然言語処理によるアプローチを用い、キーワード抽出の精度の向上を行う。その後、アプリケーション全体を通したパフォーマンスを調べるためのユーザ実験を行っていく必要がある。

### 4 関連研究

小説の作業を自動化する研究として、ストーリーの自動生成を行う研究は古くから行われてきた。BRUTUS[1]は、裏切りのような事前定義されたテーマについての短編小説を書き込むプログラムである。MINSTREL[5]は、このシステムは、保存された過去のエピソードに基づいてストーリーを作り出す。これらの自動的な小説の作成というアプローチは、作者がつまづきやすい導入部の執筆をサポートする場に役立つことが考えられる。しかし、本文の執筆は作者の特徴が最も現れる部分であり、今回のアプローチでは最低限の補助に留めるものとする。

文学的プログラミング[3]は、ドナルド・クヌースによって導入されたプログラミングへのアプローチである。我々は、文芸的プログラミングを参考として、我々のシステムにその基本的な考え方を組み込んでいく。

Spell checking[6]は、スペルミスを減少させるための基本的な方法を示している。プログラミングエディタでは、構文のハイライト表示がミスの削減のために広く用いられている。したがって、我々は、これらの機能が小説の執筆をサポートするために使用できると期待している。

### 参考文献

- [1] S. Bringsjord and D. Ferrucci. *Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine*. Psychology Press, 1999.
- [2] M. Haverbeke. Codemirror (version 2. x), 2011.
- [3] D. E. Knuth. Literate programming. *The Computer Journal*, 27(2):97–111, 1984.
- [4] T. Kudo. Mecab: Yet another part-of-speech and morphological analyzer. <http://mecab.sourceforge.net/>, 2005.
- [5] S. R. Turner. Minstrel: a computer model of creativity and storytelling. 1993.
- [6] Y. Zhao and K. Truemper. Effective spell checking by learning user behavior. *Applied Artificial Intelligence*, 13(8):725–742, 1999.