

知的バージョン管理システムに基づく論文添削支援システムの試作

A Paper Editing Supporting System Based on Intelligent Version Control

山添 久稔*¹ 大冨 忠親*¹ 新谷 虎松*¹
 Hisatoshi Yamazoe Tadachika Ozono Toramatsu Shintani

*¹ 名古屋工業大学大学院 情報工学専攻

Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

Version control systems are useful for software development. However, the systems are not appropriate for writing research papers because differences among papers are not suitable for understanding the progress of the writing. We have been developing an intelligent version control system for writing the papers and a paper editing support system based on the version control system. The system is designed to support three tasks, (1) correcting papers, (2) tracking revisions of papers, and (3) evaluating a degree of progress of writing papers. We proposed a method to evaluate a degree of the progress on the intelligent version control system.

1. はじめに

本研究では、執筆者と添削者の双方の作業をサポートすることによる、知的な論文添削支援システムの実現を目的としている。この目的を達成するため、執筆・添削途中の差分情報に基づく進捗度を用いた論文執筆システムを開発した。論文の添削作業は、通常、執筆者による論文の修正と添削者による論文の添削を複数回繰り返すことにより、進行していく。この執筆者と添削者による複数回の論文添削のやりとりを、添削サイクルと本研究では呼ぶ。添削サイクルの課題として、以下の三つが考えられる。1) 変更が行われていない文章まで閲覧を行ってしまう課題。2) 添削者が、前回の添削サイクルにおいて、自身が添削を行った部分を忘れてしまい、執筆者の修正が正しく行われたかを正しく確認できない課題。3) 論文サイクルの中で、現時点の論文の完成度がどの程度のものなのかが容易に確認できない課題。本研究では、これらの課題に対し、論文サイクルに適したシステムを試作することで、解決を試みた。試作したシステムにより、1), 2), 3) の課題に対し、執筆者と添削者双方の作業を支援を行うことが可能となった。なお、本システムを用いる環境として、大学における研究室を想定している。具体的には、1名ないし、2名程度の少数の教員が、同時並行的に複数の学生の論文添削を行っているような環境である。

2. 背景

2.1 論文執筆・添削における添削サイクル

本研究における添削サイクルは、以下に示すように進行することを想定している。はじめに、執筆者は、論文内の文章以外の付加情報のない論文を添削者に提出する。次に、添削者は、提出された論文に対し、添削をおこなう。そして、添削者は、添削を行った論文を執筆者に返却する。最後に、執筆者は、返却された論文に対し修正を行い、再度提出する。

この添削サイクルにおいて、以下の課題が存在する。1) 変更が行われていない文章まで閲覧を行ってしまう課題。2) 添削者が、前回の添削サイクルにおいて、自身が添削を行った部分を忘れてしまい、執筆者の修正が正しく行われたかを正しく確認できない課題。3) 論文サイクルの中で、現時点の論文の

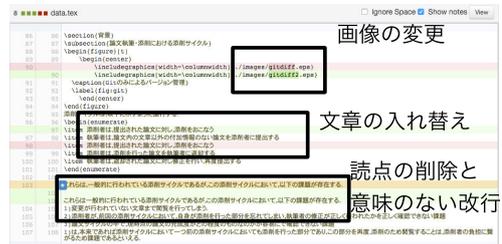


図 1: Git のみによるバージョン管理

完成度がどの程度のものなのかが容易に確認できない課題。

1) は、添削者が前回以前の添削サイクルにおいて、添削済みであり、再度添削する必要のない部分まで、閲覧を行わなければいけないことが、添削者の負担に繋がるため課題あるといえる。2) は、1) 添削者が閲覧すべき場所の正しい把握が困難であるという課題である。添削者が前回の添削サイクルにおいて添削を行った場所を想起しながら添削を行うことは、添削者の負担に繋がるため課題であるといえる。3) は、例えば、ある一つの学会に向け、複数の執筆者が同時並行的に論文の執筆を行う場合、添削者は、執筆者から提出された論文の前回の添削サイクルとの差分を基準とし、これらの論文を添削する。しかし、添削サイクルの中で、添削者は全員分の差分を瞬時に知ることは困難である。よって、提出された論文を添削に先立ち、目を通さなくてはならない。この目を通すという行為は添削者の負担に繋がるため課題であるといえる。

これら3つの課題を解決する手段として、差分情報による論文のバージョン管理を提案する。バージョン管理を行うことで、添削サイクルにおいて、一貫して添削者による添削と執筆者による修正の情報を保持と参照が可能である。一貫した情報の保持により、今回の添削における論文と、前回以前の添削情報を比較可能であり、1), 2) の課題を解決可能である。

2.2 既存のバージョン管理システムの課題

バージョン管理におけるシステムには、Apache Subversions*¹ や Git*² などが存在する。本システムでは、論文執

連絡先: 山添 久稔, 名古屋工業大学大学院 情報工学専攻, 愛知県名古屋市中区御器所町, 052-735-5584, hisatoshi@toralab.org

*¹ <https://subversion.apache.org/>

*² <https://git-scm.com/>

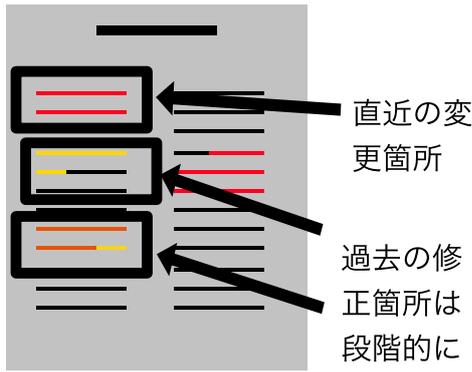


図 2: 提案する履歴可視化機能

筆を TeX で行うことを前提とし、Git を用い、論文のバージョン管理を行う。Git を用いる理由は、Git の認知度の高さである。完全に新規のシステムを導入すること比べ、システム習得における負担を軽減できると考えたためである。しかし、Git を論文のバージョン管理に適用する際、いくつかの課題が存在する。図 1 は、論文のバージョン管理に Git をそのまま用いた図である。図 1 では、以下のような課題が存在していることが見受けられる。画像の変更という大きな変更も、行の変更にとめられてしまう課題。文章の入れ替わりが起こった場合、行の削除と追加とされてしまう課題。意味のない改行も管理されてしまう課題。読点一つの修正に対し、行全体の変更と受け取ってしまう課題。これらの課題は、Git がソースコードのバージョン管理を目的としているため、発生する。これらの課題に対し、本研究では、TeX ファイル内の論文文章内にアノテーションを付加することによる解決を目指した。アノテーションによって Git の差分情報を補完することで、Git を論文に適した形へと調整を行った。

3. 知的なバージョン管理システムとその表現

知的なバージョン管理とは、論文に適した形によるバージョン管理を行いつつ、サブセクションで述べる 3 つの機能を有するものである。それぞれの機能について述べる。

3.1 変更履歴の表現

まず、1) の課題の解決を目指し、添削サイクルにおける変更履歴を色情報により可視化するシステムを実現した。添削者が見るべき論文の部分は、執筆者が論文に対して行った修正の中で、より新しい編集であるといえる。これは、執筆者が行った古い修正は、添削者がすでに目を通している可能性が高く、添削者が閲覧を行う必要が少なくと考えられるためである。またこの場合、Git のように直近の一回のとの比較による視覚化では不十分であると考えられる。これは、添削サイクルにおいて、直近に添削が行われた部分と、比較的最近添削が行われた部分は関連していることが多く、逆に、古い時点で添削者によって添削が行われ、執筆者によって修正が行われた部分は、直近の添削と関連していることがあまりないためである。よって、本システムでは、編集された時期による色情報により、変更履歴を表現する手法をとる。図 2 に示したものが提案手法である。図 2 に表すように、修正が行われた時期を添削者が一目でわかるように表す。この手法により、添削者は、論文内のどの部分を閲覧するべきかを、段階的に把握することが可能となり、添削者の負担の軽減を行うことが可能になると考えられる。

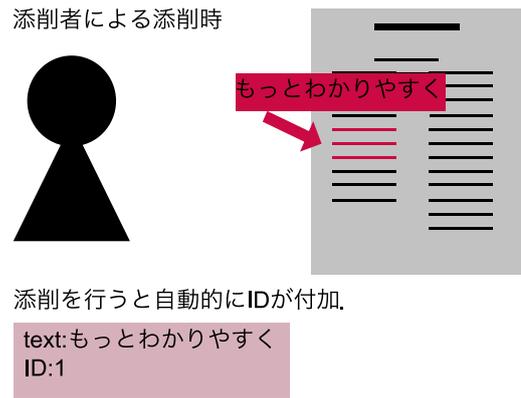


図 3: 添削時の自動 ID 付加

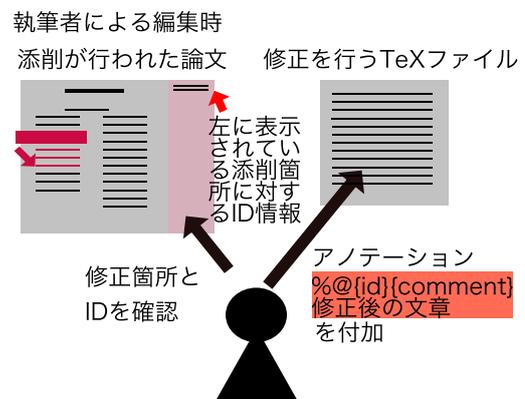


図 4: 修正時の執筆者

3.2 添削サイクルにおける修正箇所の提示

次に、2) の課題の解決を目指し、添削サイクルにおける執筆者の編集箇所をアノテーションにより、表現するシステムを実現した。2) の課題は、1) の添削箇所の把握という点では近いが、今回と前回の直近の添削に限定を行っていることに違いがある。具体的には、以下の三点である。添削者は、執筆者に対し、どのような添削を行ったか。添削に対し、執筆者はどのような修正を行ったか。その添削は、今回提出された論文の修正箇所のどの部分が対応しているか。これらの三点をシステムにより、添削者に対して、わかりやすい形で提示することで、添削者の負担軽減に繋がるといえる。

そこで、図 3~5 のような添削システムを提案する。まず、図 3 は、添削者が執筆者により提出された論文に対して、論文の添削を行っている様子を表している。簡単のため、今までに添削情報がない初めての添削を行っている様子を表している。添削者は、添削箇所を選択した後、選択箇所に対してコメントを付加することで、添削を行う。この際、各添削に対して、システムが ID の付加を自動で行う。そして、図 4 は添削者による添削を受けた論文に対し、執筆者が修正を行っている様子である。この際、執筆者は、各添削の内容を確認すると同時に、その添削に付加された ID を確認する。そして、TeX ファイルを修正する際、この ID をコメントとともにアノテーションとして、TeX ファイルに記述する。アノテーションは、図 4 に示すように、`%@[id]{comment}` と記述する。添削に自動付加された id と、この修正に対するコメントを `%@` に続く波括弧に

添削者による修正の確認

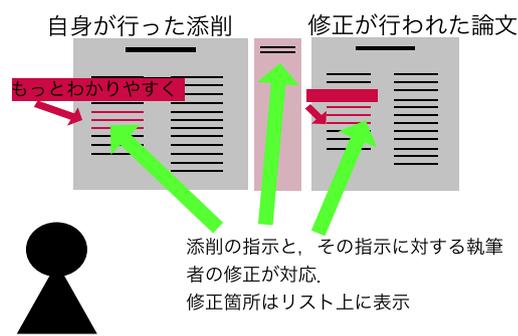


図 5: 添削者による修正箇所の確認

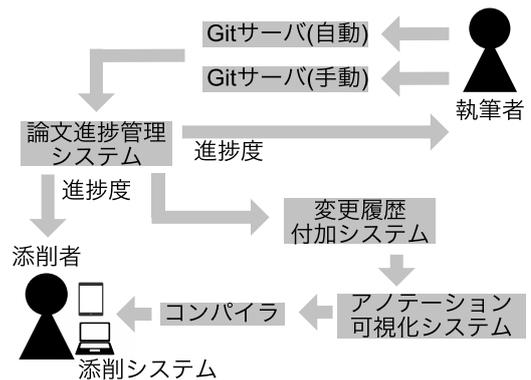


図 6: システム構成図

句点の数, 読点の数, セクションの数, サブセクションの数, 段落数, 段落における平均行数, 引用論文数, 論文内の画像数, 論文内の参照数, 箇条書きされている文章数, 執筆者による修正によって変更した文字数, 論文内に存在している名詞の数, 添削サイクルにおける添削数, git の履歴の間隔, 論文全体の文章量 タイトルに含まれる名詞の登場数
--

表 1: 式 (1) で用いるパラメータ x_i の例

それぞれ記述する。修正後の文章は、アノテーションを記述したのち、改行を行い、その後記述する。この操作により、TeX ファイルに添削者による添削情報と、その添削に対する執筆者の修正情報・コメントを付加させる。最後に、図 5 は、執筆者の修正を受け、再度添削者に提出された論文を添削者が閲覧を行っているモード図である。添削者は、自身が行った論文の添削情報を参照しながら、執筆者の提出した修正を閲覧を行う。また、中央に、添削を行った箇所がリスト状に表示されており、このリストを操作することにより、対応する修正箇所にジャンプすることが可能である。この機能により、添削者は、一目で執筆者による、添削者の添削に対する修正を閲覧することが可能となり、添削者の負担の軽減に繋がると考える。

3.3 進捗度を用いた論文進捗管理システム

3) の課題を解決することを目指し、進捗度を導入した論文進捗管理システムを実現した。論文の進捗を経験的に表しているいくつかの要素の線形和を用い、その論文が過去の添削サイクルにおける論文より、完成に向け、どの程度進捗しているかを表現することを目標としている。

論文の進捗を定量化する際に、単純な分量による定量化は困難である。例えば、論文における文章量に着目する。論文の完成には、一定量の文章を作成することが必要であり、論文の文章量は進捗度の算出に影響していると考えられる。しかし、文章を推敲する際に、文章量が減ることもある。この際、本来であれば、論文の完成に向け、進捗が存在するが、単純に文章量のみに着目した場合正しく、進捗度を算出することができない。そこで、本研究の進捗度の算出には、添削サイクルにおいて収集可能な情報を用いて、機械学習による、ヒューリスティックな進捗度の算出を目指した。具体的には、進捗

度の算出に用いる要素は、表 1 の通りである。

表 1 に列挙した、要素の集合を A として、進捗度の算出に用いたモデルを、式 (1) に示す。

$$\text{進捗度} = w_0 + \sum_{i \in A} w_i x_i \quad (1)$$

x_i は表 1 に列挙した要素の値が入る。 w_i は、 x_i に対する重みである。 x_i の各値について述べる。句点の数, 読点の数, セクションの数, サブセクションの数, 段落数, 論文内の画像数, 論文内の参照数, 箇条書きされている文章数, 論文内に存在している名詞の数, タイトルに含まれる名詞が論文中の登場数とは、論文中に登場するそれぞれの数のことである。引用論文数とは、論文において、引用が行われている論文の数である。段落における平均行数とは、論文中の各段落における文の数の平均である。執筆者による修正によって変更した文字数とは、添削者の修正指示に対し、執筆者が今回の論文サイクルにおいて修正し、書き直しを行った文字数のことである。なお、この文字数とは添削サイクルにおける前回は執筆者が添削者に提出した論文と、今回執筆者が添削者に提出した論文との編集距離のことである。添削サイクルにおける添削数とは、前回の執筆者の論文の提出に対し、添削者が添削を行った数のことである。Git の履歴の間隔とは、添削者の添削を受け、返却された論文に対し、提出者が再度提出を行うまでの時間的感覚のことである。

4. 実装

4.1 システム構成図

これまでの提案手法を用いた知的な論文バージョン管理システムの構成図を図 6 に載せる。執筆者は、まず、Git サーバに論文のリポジトリを作成し、作成したリポジトリを論文進捗管理システムに登録する。このシステムとは、サブセクション 3.3 で述べた複数人に対応した論文進捗管理システムである。本システムでは、一つの論文執筆に際し、二つのリモートリポジトリを用いる。Git 執筆者が、執筆のキリがついた等の任意のタイミングで *gitpush* を行うリポジトリと、システムにより一定時間毎に自動で *gitpush* を行うリポジトリとである。これらの二つのリポジトリを用い、論文進捗管理システムは、現在の論文の進捗度を随時算出し、添削者と執筆者に進捗度を提示する。

添削者が論文の添削を行う際、取り出された論文は、変更履歴付加システムとアノテーション可視化システムを通り、コ



図 7: 論文進捗管理画面



図 8: 添削画面

ンパイラによって PDF ファイルに変換された後、添削システムを通じて添削者に提示される。変更履歴付加システムとは、節 3.2 で述べた変更履歴の色情報による可視化のことである。論文進捗管理システム上に存在する添削サイクルにおける過去の情報をもとに、変更履歴付加システム上で色情報の付加を行う。色情報の付加は、TeX ファイルを変更履歴付加システムにより自動的に編集することでおこなう。

アノテーション可視化システムとは、節 3.1 で述べた添削における自動添削箇所への提示と執筆者による明示的な修正箇所への提示を行うシステムである。変更履歴付加システムと同様に、アノテーションを元に自動的に TeX ファイルの編集を行う。また、編集を行う際、論文進捗管理システムより添削者による添削情報を取得する。

最後に、TeX ファイルは、コンパイラによって、PDF ファイルに変換され、添削システムに渡される。添削システムは、添削者の PC およびタブレット端末 (iPad^{*3}) 上で動作する。添削者は、環境を選ばずに添削を行うことが可能である。節 4.2 にて、添削システムの実行例を紹介する。

4.2 実行例

図 7 で表されるのは、図 6 で示した論文進捗管理システムである。図 8 で表されるのは、図 6 で示した添削システムである。論文進捗管理システムでは、各執筆者の論文のサムネイルが進捗度、差分情報とともに表示される。直近の大まかな進捗度は、執筆者の名前右横に矢印の傾きとして表現される。この矢印により、添削者は執筆者の執筆状況を知ることが可能となる。また、表示されるサムネイルの差分情報は、色情報として表現される。この差分情報は、現在の最新の論文と、過去の

論文の中で進捗度がある一定の閾値を上回る最新のものととの差分情報の表示を行う。また、その論文の提出された日時と、現在と比較していくつ古いバージョンであるかの表示を行う。これにより、添削者は、執筆者の執筆ペースに合わせた差分情報の閲覧を行うことが可能となる。

添削サイクルにおいて、添削者が、執筆者によって提出された論文の添削を行うシステムである。添削システムの画面は大きく分けて、前回の添削情報、添削に対する修正リスト、執筆者の修正の三つに分けることができる。それぞれについて、述べる。

前回の添削情報の画面は、図 8 左側の画面である。前回の添削者が行った添削が表示されている。“リストが対応”と書かれた矢印が指し示す場所に文字列が存在する。これは、前回の添削において、添削者が添削を行った部分である。添削者は、添削者が前回の添削サイクルにおいて行った添削を、修正された論文を閲覧する際に参照することができる。添削に対する修正リストの画面は、図 8 中央の画面である。添削者が前回行った添削の一覧の表示が行われる。一覧に表示されている各添削をマウスでクリックすることにより、対応する修正の箇所までジャンプする。これにより、添削者の修正箇所を探す負担を軽減した。執筆者の修正の画面は、図 8 右側の画面である。執筆者によって、修正された論文の表示が行われる。添削者は、前回添削サイクルにおける添削を参照しながら、添削を行うことが可能である。

また、“色情報による変更履歴の表現”と矢印が論文内の文章を指し示している。これは、色情報による編集履歴を表している。色が変更されている箇所は、前回の添削サイクルにて、添削者による添削が行われていない。これは、前回の添削サイクルにて、添削が行われていない箇所は、論文の部分的な完成に近づいていると考えるためである。色が変更されていない箇所は、前回の添削サイクルにて、添削者による添削が行われている。これは、前回の添削サイクルにおいても、添削が行われている箇所は、論文の部分的な完成にあまり近づいていないと考えるためである。節 3.1 で提案したシステムにより、色が決定され、論文のどの部分の完成度が低いか、どの部分の完成度が高いかを一目で添削者に提示することで、添削者の負担の軽減に繋がると考えることができる。

5. おわりに

本研究では、執筆者と添削者の双方の作業をサポートすることによる、知的な論文添削支援システムの試作を行った。論文添削における課題を、変更が行われていない文章まで閲覧を行ってしまう課題。添削者が、前回の添削サイクルにおいて、自身が添削を行った部分を忘れてしまい、執筆者の修正が正しく行われたかを正しく確認できない課題。論文サイクルの中で、現時点の論文の進捗がどの程度のものなのかが容易に確認できない課題。の三つであるとして、それぞれの課題に対しての解決方法を提案した。

参考文献

- [片山 14] 片山 真也, 合田 拓史, 白松 俊, 大園 忠親, 新谷 虎松: バージョン管理を考慮した Web に基づくリアルタイム協調型論文作成支援システムの実現, 情報処理学会第 76 回全国大会 (2014)

*3 <http://www.apple.com/jp/ipad/>