

リアルタイム遠隔文章作成指導支援システムの開発

Development of Realtime Remote Collaborative Writing Support System

山岸 芳夫^{*1}, 稲置 美里^{*1}Yoshio Yamagishi^{*1}, Misato Inaoki^{*1}^{*1}金沢工業大学 情報フロンティア学部 メディア情報学科^{*1}Department of Media-Informatics, College of Informatics and Human communication, Kanazawa Institute of Technology

Email: yamagisi@neptune.kanazawa-it.ac.jp

あらまし: 小論文や履歴書などの文章作成を指導する場合、リアルタイムに文章作成者と会話しながら行った方が能率が良いが、指導者が離れた場所にいる場合はこのようなことは困難である。我々はWebSocketを用いて、リアルタイムの遠隔文書作成指導が可能なシステムを構築した。完成後本システムのユーザビリティ評価を行い、概ね見やすく分かりやすいが反応や信頼性に若干問題がある、という結果を得た。

キーワード: ディスカッション支援, CSCL, Node.js, 文章作成支援

1. はじめに

大学生にとって文章作成能力は、卒業研究論文や就職活動のエントリーシート、小論文の出来栄に直結するため、在学中に磨いておくべき必須のスキルの一つと言える。そのため、文章作成能力の指導に注力する大学も多い。金沢工業大学（以下本学）には、学生の書いた文章を添削、指導して文章作成能力の向上を図る「ライティングセンター」が存在している。ここではスタッフが学生に対してヒアリングを行い、用いるべき語彙や言い回しについてアドバイスしながら文章を添削していく。しかし、このような懇切丁寧なサービスが提供されているにもかかわらず、ライティングセンターにまで足を伸ばすことを負担に感じる学生が多く、利用者が増加していないのが実情である。メールで添削も行えるが、指導者と作成者が互いの意思を確認しながら文章の添削を行うためには、何度もメールのやり取りをしなければならず、時間がかかってしまう。そこで本研究では、ライティングセンターの利便性を高めるため、遠隔で添削支援が行えるシステムを開発した。

2. 開発および実行環境

本システムはNode.js⁽¹⁾を用いたWebアプリケーションである。Node.jsはサーバ側のプロセスとブラウザとの双方向通信が可能で、リアルタイムに複数のブラウザ間の通信を実現するシステムである。Node.jsは通常WebSocketを用いて双方向通信を行うが、ブラウザが未対応の場合もあり、必要に応じてポーリングなどの従来技術も用いている。

本システムの開発においては、Node.jsで作られたタスクランナーの1種であるgulp.js⁽²⁾を利用し、CSSの圧縮、JavaScriptの難読化、Node.jsサーバの立ち上げを自動化させた。

3. システムの概要

本システムでは、ユーザ認証にGoogle OAuth2.0を利用しており、Googleにアカウントがあれば誰でも利用が可能である。本学の学生であれば全員Gmailのアカウントを所有しているため、何も登録せずに使用することが出来る。

ユーザーがログインが完了すると、ルーム選択画面に移行する。この画面では「既に作成されているルームへ入室する」と、「新しくルームを作成し、入室する」のいずれかを選択して入室する。

ルームを選択すると添削画面(図1)に移行する。添削画面には主に、「テキストチャット機能」「文章アノテーション機能」がある。添削文章はプレインテキストに限られ、新規の場合はコピーペーストもしくは書き下ろしで文章を添削画面に挿入する必要がある。

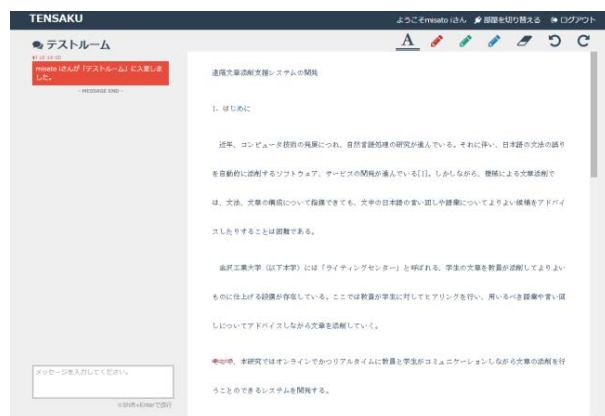


図1: 本システムの添削画面

添削画面では、過去に添削履歴のあるルームに入室した場合、チャットメッセージ、添削文章、添削文章のアノテーションが復元される。画面左側はチ

チャット画面になっている。ユーザが送信したメッセージの他に、ルームメンバーの入退室も通知される。本システムで利用できる添削機能は、文字色および背景色の変更、二重取り消し線の追加、フリーハンドで図形が書けるペンツールである（図 2）。

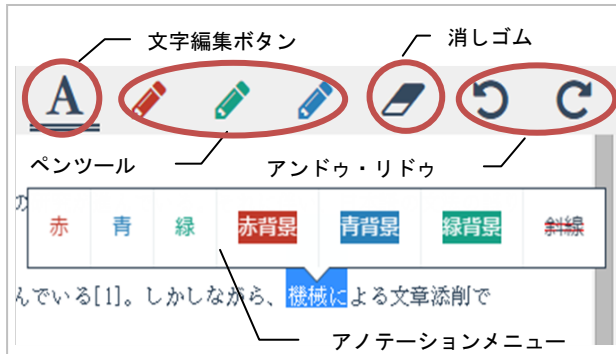


図 2: 本システムの添削機能

文字編集ボタンをクリックすると文字編集モードになり、このモード時に任意の文字列を選択すると、文字列の上にメニューが表示され、追加したいアノテーションを選択できる。消しゴムはペンツールで描いた軌跡を消すための機能で、アンドウ・リドゥは添削の取り消し、やり直しを行う。

4. システムの評価

本システムの完成後、以下の様にユーザビリティの評価を行った。

本学学生 10 名を 2 人 1 組のペアに分け、それぞれで「教師役」「生徒役」を決め、前半、後半の二回に分けて本システムを利用させた。前半、生徒役の用意した文章を教師役の学生に添削させ、完了後にアンケートを行い、後半に役を交代し同様に添削、アンケートを行なった。アンケートはウェブユーザビリティ評価スケール⁽³⁾を参考に設問を設定した。

教師役の結果を図 3 に示す。教師役では「好感度」以外の評価が前半組に比べて後半組が低い事が分かる。生徒役の結果を図 4 に示す。生徒役の方は、「好感度」「構成の分かりやすさ」を除いて結果に差は見られなかった。両図とも「好感度」は前半に比べて後半の評価が高くなったことがわかる。

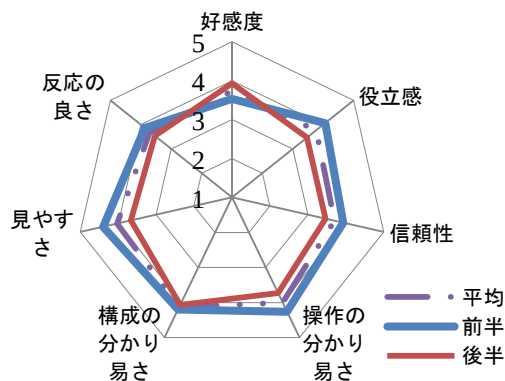


図 3: 「教師役」のアンケート結果

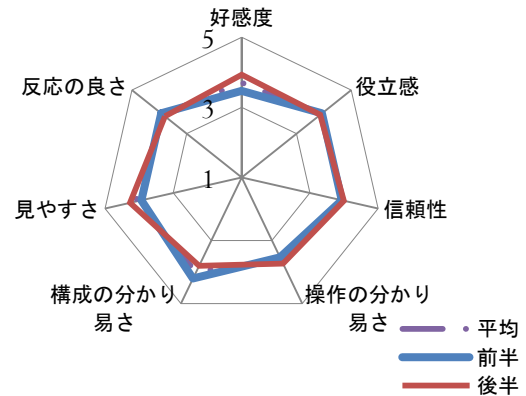


図 4: 「生徒役」のアンケート結果

前後半を合わせると、教室役と生徒役の評価はともに平均点 3 より高く、総合計の差は 0.02 ポイントでほとんど差はなかった。しかし、「反応の良さ」「信頼性」の評価は教師役が生徒役の評価に比べて低く、「操作の分かり易さ」は生徒役の評価が低かった。

5. まとめと今後の課題

まず、教師役において前半組に比べ、後半組の評価が全般的に低かった原因については、前半組は本システムのアノテーション機能やチャット機能に初めて触れ、印象が強かったが、教師画面と生徒画面の機能的な違いはなく、後半組は画面が同じことに若干期待外れな思いを抱いたため、と考えられる。

次に、教師役の「反応の良さ」の評価が低い原因は、試行中しばしば発生したペンツールの動作の遅延と考えられる。また、「信頼性」が低い原因については、アンケートの自由記述の中に、生徒役と教師役が同時に編集を行った場合に正しく処理できていなかった、という指摘があり、コリジョン処理に不具合があったと思われる。

生徒役の「操作の分かり易さ」の評価が低いのは、ルーム選択画面の使いづらさや、文字編集モードとペンツールの使い分けの分かりにくさが原因と考えられる。

今後はコリジョン処理を正しく行えるように、互いのユーザの処理の履歴を残す機能の追加やインターフェースの改善を行う。さらに、現状では一つのルームに 2 名までしか入れないが、2 名以上の文章作成、添削の実現も今後は検討したいと考えている。

参考文献

- (1) Node.js, <http://nodejs.org/>.
- (2) gulp.js, <http://gulpjs.com/>
- (3) 仲川薫他, “ウェブサイトユーザビリティアンケート評価手法の開発”, 第 10 回ヒューマンインターフェース学会 研究紀要 (2001)