

Chatplexer: チャットを併用する口頭発表 における重要発言選択支援の試み

小林 智也^{†1} 西本 一志^{†2}

近年、チャットなどの短いテキストを即時交換できるメディアを対面口頭での発表・質疑と平行して利用する試みが増えている。対面口頭対話とは異なりチャットには話者交代などの制限がなく、より広い視点からの意見をより多く議論に取り込む目的で行われている。しかし発表者が発表中にもチャットに注意を払い続けることは難しく、発表者が重要だと思うチャット発言を議論に取り上げることは難しい。本論文では、チャットから対面口頭対話に対して返信することのできる「クロスチャンネル返信(XCR)」を提案し、XCRを分析することによって重要なチャット発言を自動的に学習・推定することを試みた。XCRを実装したChatplexerシステムを使用して実験したところ、チャット上の発言の過半数はXCRに対する返信とその子孫ノードであり、重要な発言もそれらのチャット発言であることが多いことが分かった。また、XCRの情報をを用いると、重要な発言をJ4.8で学習・推定させた場合に適合率が大きく改善することが分かった。

Chatplexer: Supporting extraction of important opinions in an oral presentation where a text-chat is concurrently used

TOMOYA KOBAYASHI and KAZUSHI NISHIMOTO

Recently, there have been many attempts that use short-text-exchanging media concurrently like a chat in a face-to-face meeting in order to obtain various opinions from wider viewpoints by more audience. Such media are free from some restricts of the face-to-face meeting, e.g. a rule of turn taking. However, it is actually difficult for a presenter to pay attention on the chat while he/she is presenting. He/she cannot discuss on important chat opinions in his/her presentation. In this paper, we propose a “cross-channel reply” (XCR) that allows the audience to reply the contents of the face-to-face meeting channel from the chat channel. By analyzing the XCRs, we attempt to automatically estimate the important chat opinions using J4.8 decision tree. We implement a novel chat system named “Chatplexer,” which is equipped with XCR function. We

conducted user studies using Chatplexer. As a result, we found that more than half of the important chat opinions belong to trees whose root nodes are XCR messages, and that using XCR information the precision ratio is improved.

1. はじめに

近年、対面口頭での発表や議論と並行して、チャットやTwitterなどの短いテキストを即時交換できるメディアを使用する試みが多数行われている（以下では、このような形態の会議を総称して「チャット併用会議」と呼ぶ）。チャット併用会議の最初の例は、暦本らによるWISS '97^{*1}でのComicChatを用いた実験である¹⁾。以後WISSでは“WISS Challenge”²⁾として、対面口頭対話による会議を支援するチャットなどのコミュニケーション・システムを募集し、ワークショップの開催中にこれらのシステムを実際に併用することを継続的に試みている。

このようなWISSでの取り組みに触発され、通常のチャットを用いたチャット併用会議の実験が国内外で多数実施されてきた¹⁾³⁾⁴⁾。また、授業でチャットを併用する実験も多数行われている⁵⁾⁶⁾。

こうした試みは、より広い視点からの意見をより多く議論に取り込むことを期待して行われている¹⁾。対面口頭対話では発言権は排他的であり、1人ずつ順番に発言することを強いられる。このため、時間的制約によって質疑や意見を述べる機会は極めて限られた数の聴衆にしか与えられない。これに対してチャットでは発言権は排他的でなく、複数の参加者が同時に発言することが可能であるため、いつでも自由に発言できる。さらに、チャットは対面口頭対話に対しても独立しているため、口頭で発表中あるいは質疑応答中でも、関係なく発言することができる。従って、口頭で質問することができない大多数の聴衆にチャットという発言チャンネルを与えることによって、発言の数を増加させることが期待できる¹⁾。実際にいずれの研究においても、チャットは活発に利用され、参加者からも非常に有用であったという意見を得たと報告されている。

^{†1} 北陸先端科学技術大学院大学 知識科学研究科

Knowledge Science, Japan Advanced Institute of Science and Technology

^{†2} 北陸先端科学技術大学院大学 ライフスタイルデザイン研究センター

Research Center for Lifestyle Design, Japan Advanced Institute of Science and Technology

*1 日本ソフトウェア科学会主催 Workshop on Interactive Systems and Software 1997.

一方、特に学会発表のような発表者と聴衆とが明確に分かれている形式の会議（以下ではこの形式の会議を「プレゼン型会議」と呼ぶ）においては、チャット併用によって聴衆から提出される意見が増える反面、発表者がチャット上での意見や議論を把握することが困難であることが指摘されている⁵⁾。これは発表者にとって、対面口頭での発表や質疑にもチャットにも同時に注意を払い続けることの認知負荷が非常に高いためであると考えられる。会議終了後に十分な時間をかけてチャットの発言ログを吟味すれば、チャット上での議論内容ある程度把握することはできるだろう。しかし、発表中や質疑応答中に発表者がチャット上の重要な意見や質問に気づき、それらをその場で取りあげて会議参加者全員で議論することができれば、より望ましいと考える。

我々は、プレゼン型会議の中でも、講義やセミナーのように講師が聴衆に対して知識を教授・伝達することを主目的とするタイプの発表（知識伝達型プレゼン）ではなく、学会発表のように発表者が新奇な仮説や知見を発表し、これに対する聴衆からの様々な意見を収集することを主目的とするタイプの発表（意見収集型プレゼン）を対象として、発表者がチャット上から重要な意見を簡便にピックアップすることを可能とするチャット・メディアであるChatplexerの構築を進めている。これにより、発表者がチャットから取り残される問題を解消することを目指している。

本論文では、発表中ならびに質疑応答中におけるチャット・ログから、重要な意見を質疑応答中にピックアップ可能とする手法を提案し、その有効性を評価する。なお、意見収集型プレゼンでは、収集する意見は特に発表者にとって重要であることが求められる。ゆえに、チャット上の意見の「重要さ」を判断する基準は、聴衆にではなく発表者に置くこととする。

以下、2章では関連する研究について紹介する。3章では、発表者にとって重要な発言を抽出するための方法を提案する。4章では、提案した手法に基づき実装したシステムの構成について説明する。5章では、実装したシステムを用いた実験について述べ、6章では実験結果を示し、7章では実験結果に基づき提案手法の有効性を議論する。8章はまとめである。

2. 関連研究

従来から、チャット併用会議においてチャット・ログから重要な発言を投票によって抽出する試みはいくつかなされている。backchan.nl⁷⁾やOn-Air-Forum⁸⁾、勅使河原らの研究⁹⁾では、各チャット発言に対して聴衆が投票を行う機能を提供している。backchan.nlでは多数の賛同票を集めた記事だけを選ぶことで、発表者が見るべき発言の数を減らして支援している。しかし、これらの試みでは重要性の判断を聴衆が行っているため、必ずしも発表者

にとって重要な発言が抽出されるとは限らない。例えばBackchan.nlでは、「お腹がすいた」などのその時々聴衆の感情を表す発言がランキングの上位になることがあったと報告されている。

チャット併用会議での使用を意図したものではないが、由井蘭ら¹⁰⁾は、チャットの各発言に対して「質問」「回答」などの意味タグを付加するセマンティックチャットを提案し、これによって各発言の意図を明確にする試みを行っている。しかし、この試みにおいても発言の意味タグを付与するのは各発言の発言者であるため、これをプレゼン型会議に併用した場合、やはり重要性等の判断は聴衆が行うことになる。

On-Air-Forum⁸⁾においては、スライドのページと座標を指定してコメントを送信することができる。つまり、各コメントはスライド上に記述されているコンテンツと密接にリンクするものとなる。しかし、スライド上にすべての発表内容が記載されているわけではないため、口頭による発言内容に対してもコメントできることが必要である。

以上のように、従来のチャット併用会議におけるチャット上の発言の重要性は聴衆が判断するものばかりであった。また、発表者が口頭で話した内容に対してコメントを明示的に付与する手段を提供する試みは、我々の知る限り存在しない。本論文では、次節で提案する手法により、スライドのみならず発表者が口頭で話した内容に対してもコメントを明示的に付与することを可能とし、さらにこの返信関係情報をもとに、チャット上の発言の重要性を発表者の基準で推定することを目指す。

3. 提案手法

McCarthy らは、チャット併用会議を観察・分析した結果、対面口頭対話と並行して行われるチャットは、バックチャンネルに類似しており、対面口頭での議論から時間的・意味的制約を受けると考察している¹¹⁾。この知見に基づき、筆者らは、チャット併用会議においては対面口頭での対話内容とチャット発言との間の対応関係を明確化し、これを記録することが重要であると考えた。さらに発表者の口頭での発言に対する返信であれば、それは発表者にとっても重要である可能性が高いと考えられることから、この対応関係を記録し分析を行うことによって、発表者にとって重要なチャット発言を推定できると考えた。この分析によって、発表者にとって重要な発言を抽出するアルゴリズムを実現できれば、時間的制約がある質疑時間においても、チャット上の有益な意見を取り入れて議論を進めることが容易になると期待される。

そこで本論文では、対面口頭での対話内容とチャット発言との間の対応関係を明確化し、

記録するための手段として、「クロスチャンネル返信」という概念を提案する。チャット併用会議において用いられるメディアは、音声（口頭）、スライド、チャットの3つである。音声とスライドは対面口頭対話として1つのチャンネルを形成し、会議の主となる話題が扱われるフロントチャンネルになる。これに対してチャットは、単体で1つのチャンネルを形成し、バックチャンネルとして機能する。クロスチャンネル返信は、フロントチャンネルの対面口頭対話の内容に関連づけて、バックチャンネルのチャットから返信する機能である。具体的には「口頭で発言中の内容に対する返信」（以後、**対口頭対話返信**と呼ぶ）と、「スライドに対する返信」（以後、**対スライド返信**と呼ぶ）の2種類の返信機能を提供する。

例えば、発表者が現在口頭で説明中の内容に対して何らかの疑問や意見を持った際には、口頭で発言中の内容に対する返信としてチャット発言を送信する（対口頭対話返信）。また、発表者がこれまでに表示したプレゼンテーションのスライドに対して何らかの疑問や意見を持った際には、スライドの内容に対する返信としてチャット発言を送信する（対スライド返信）。この2種類のクロスチャンネル返信によって、バックチャンネルからフロントチャンネルへ、チャンネルを越えた明示的な返信が可能になる。この関係を以下の図1に示す。

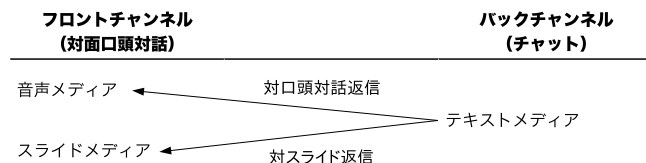


図1 メディアおよびチャンネルと返信の関係

なお、クロスチャンネル返信としては、対面口頭対話（フロントチャンネル）からチャット（バックチャンネル）への返信も考えることができる。これについては、発表者が発表者自身の観点によってチャット上から重要な発言を採り上げることができないという、本論文で取り扱う問題が解決したあとに検討する予定としているため、本論文では対象としない。

4. システムの構成

クロスチャンネル返信機能を実装したチャット・システム「Chatplexer」を開発した。Chatplexerは、新規発言と通常の返信に加え、クロスチャンネル返信を入力・保存及び表示することができる。システムの全体像を図2に示す。図中の「スライド発言」については

「発表者用クライアント」の項で説明する。

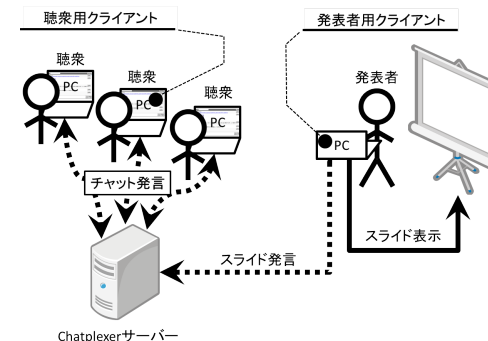


図2 システムの全体像

4.1 サーバ

サーバはPHPモジュール（5.x）を組み込んだApache HTTPサーバ（2.2.x）、MySQLデータベースサーバ（5.x）によって構築されている。サーバに用いた各ソフトウェアは適時最新バージョンに更新された。後述する聴衆用システムで低遅延でのメッセージ配信を実現するためLong-Polling技術を使用するので、Apache及びMySQLは多数の接続を高速に受け付けるよう設定を調整した。

Chatplexerでは、聴衆が投稿する通常の文字列での発言（チャット発言）の他に、後述するように発表者が現在表示しているスライドを示すスライド発言も扱う。サーバは、これらの発言に対して正の整数による一意のシーケンス番号を割り当ててデータベースに記録する。シーケンス番号は、最初の発言が1であり、発言が投稿される毎に2、3とカウントアップする。

4.2 発表者用クライアント

発表者用クライアント（図3）は、発表者がスライドを表示して発表を行う際に用いるアプリケーションである。発表者用クライアントは、.Net Framework 2.0 Windows フォームアプリケーションとしてC#で実装した。スライドはPortable Document Format形式（PDF形式）で用意するものとした。発表者用クライアントが利用者（発表者）に対して提供する機能は、スライド用PDFを取り込んで画像に変換する機能、スライド（ページ）を全画面表示する機能、スライド（ページ）を切り替える機能、の3つである。

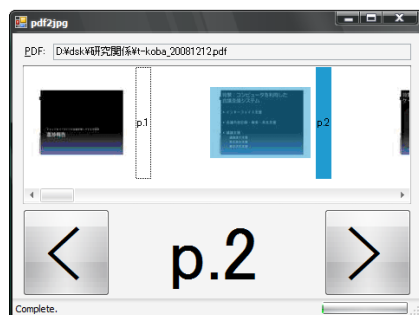


図 3 発表者用クライアントの外観

発表者用クライアントは、発表を開始するとスライドを画像に変換して全画面表示する。スライドの切り替え時には、これから表示しようとするスライドの画像が既にサーバにアップロード済みであるか（すなわち、チャット・ログ上に一度でも表示されているかどうか）どうかを確認する。アップロード済みである場合は、スライドの ID のみをサーバに送信し、ページが切り替わったことだけを通知する。アップロードされていない場合は、スライドを画像に変換したファイルをサーバにアップロードし、その後ページの切り替えをサーバに通知する。

スライドはオープンソースの PDF 描画ソフトウェアにより、JPEG 形式の画像に変換している。サーバは、発表者用クライアントからスライド切り替え通知を受け取ると、対応するスライド画像をサムネイル化したものをスライド発言として聴衆用クライアントに配信する。すなわちスライド発言とは、スライドの切り替えをトリガーとして発言される、そのとき表示されているスライドの画像を、チャット発言として表示したものである。（実際の表示例は聴衆用クライアントの項で示す）

なお、PDF 描画ソフトウェアについては、当初は Ghostscript を使用していたが、途中から mupdf に変更している。理由は、Ghostscript が一部の PDF を正常に処理できない不具合が発生したためである。

4.3 聴衆用クライアント

聴衆用クライアント（図 4）は、聴衆がチャットの表示と投稿に使用するアプリケーションである。聴衆用クライアントはウェブアプリケーションとして PHP と JavaScript で実装した。Long-Polling 技術と Ajax 技術を使用しており、発言の配信遅延は 1 秒以内である。聴衆はブラウザに聴衆用クライアントの URL を入力することによって利用する。



図 4 チャット画面

チャット・ログには、聴衆が送信する発言と、発表者用クライアントから自動投稿される、現在発表中のスライドのサムネイル画像を発言本文とするスライド発言が表示される。いずれの発言についても、シーケンス番号と送信者名が併せて表示される。ただしスライド発言の送信者名は「発表者」となる。新しい発言が追加されるとチャットは自動的に新しい発言までスクロールする。ただし発言入力欄に文字列を入力中である場合は、チャット・ログの表示範囲が変更されることを避けるため、入力中の聴衆用クライアント上では自動スクロールしない。

聴衆用クライアントからの発言送信方法としては、新規発言、通常返信、対スライド返信、対口頭対話返信の 4 通りがある。いずれの方法についても、本文や送信者名が空欄であると警告が表示され、チャット発言を投稿することはできない。また、間違えて送信してしまうことを避けるために、キーボード操作（主に Enter キー）による投稿はできないようになっている。なお、対口頭対話返信などは発言の送信時刻よりも発言の入力開始時刻の方が重要であると考えられる。それを考慮し、全てのチャット発言について、発言本文を入力し始めてから送信されるまでに経過した秒数を取得し、発言送信時刻と併せてサーバーに蓄積している。ただしこれらの送信時刻や入力にかかった時間などの時刻に関する情報は、聴衆用クライアントの画面上には表示しない。また、通常返信、対スライド返信、対口頭対話返信のいずれについても、1 つの返信発言中に含むことのできる返信タグ（後述）は 1 種類・1 つのみに制限した。

以下、各発言の送信方法について詳述する。

4.3.1 新規発言

新規発言は、先行するいずれの発言も参照しない、新たな話題の最初の発言を送信する際に使用する。一般的なチャットであればこのような発言も多数なされうるが、本論文が対象としているプレゼン型会議で並行して行われるチャットでは、このような発言が生じる頻度は少ないと予想される。このため、図4に示すように、インターフェイスの設計においては新規発言ボタンを発言入力欄から遠いところに配置した。

なお、後述する通常返信やスライド返信で使用する返信タグ「>> n」や、対口頭対話返信で使用する返信タグ「>> *」を、新規発言に手動で入力して返信発言としてしまうことを防止するために、これらのタグが含まれていると警告を表示し、投稿できない仕様とした。

4.3.2 通常返信

図4に示すように、チャット・ログ中の各発言の表示部の右端に「返信ボタン」が表示されている。入力欄に発言内容を入力した後、返信したい発言に付与されている返信ボタンを押すことで、その発言の返信として発言が送信される。送信された通常返信発言の末尾には、返信タグ「>> n」（nは返信対象発言のシーケンス番号）が自動的に付加される。この返信タグには、返信対象発言へのリンクが自動設定されているので、このタグ上にマウスカーソルを重ねることで、返信対象発言を遡ってトレースしていくことができる。

4.3.3 対スライド返信

発表者用クライアントから自動投稿される、スライドのサムネイル画像を発言内容とするスライド発言にも、聴衆からの発言と同様にシーケンス番号が付与され、さらに「返信ボタン」も付与されている。図5に、スライド発言の例を示す。

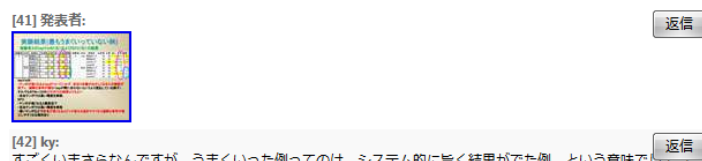


図5 スライド発言の例

チャット・ログ上にはすべてのスライド発言が残っているので、過去にさかのぼって任意のスライドに返信することも可能である。入力欄に発言内容を入力した後、返信したいスライド発言に付与されている返信ボタンを押すことで、そのスライド発言の返信として発言が送信される。送信された対スライド返信発言の末尾には、返信タグ「>> n」（nは返

信対象であるスライド発言のシーケンス番号）が自動的に付加される。この返信タグには、返信対象発言へのリンクが自動設定されているので、このタグ上にマウスカーソルを重ねることで、返信対象であるスライド発言を遡ってトレースすることができる。

4.3.4 対口頭対話返信

対口頭対話返信は、チャット上には存在しない、口頭での対話内容に対する返信である。口頭対話は音声データであるため、過去の発言内容について厳密に返信先を指定させることは難しい。そこで Chatplexer では「今喋っている内容」だけに限定して対口頭対話返信できるものとして、対口頭対話返信の入力が負担にならないようにした。対口頭対話返信の発言方法は、発言入力欄に返信内容を入力した後、発言入力欄のすぐ右にある「今の口頭会話にツッコミ」ボタンを押すだけである。対口頭対話返信を表すタグとしては、通常返信タグを変形した「>> *」という表記を用いた（図6）。この「>> *」を対口頭対話返信タグと呼ぶことにする。対口頭対話返信では、入力が終了して送信した時刻ではなく、チャット入力欄に入力を始めた時刻に対口頭対話返信を行ったと見なす。

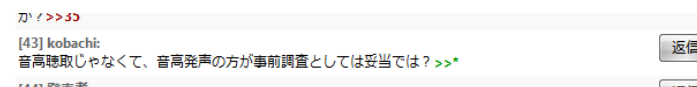


図6 口頭発言への返信の例

5. 実験

実験ではまず、Chatplexer を使用して意見収集型プレゼンを実施し、クロスチャンネル返信のデータを収集する。発表終了後、発表者に対して Chatplexer 上で行われたチャットのログを提示し、自分にとって重要と判断される発言をピックアップしてもらう。こうして得られたクロスチャンネル返信データと重要発言のデータを用いて、機械学習によって重要発言の推定を試み、その性能を検証する。

5.1 実験の手順

Chatplexer を2010年の11月～12月の2ヶ月間にわたって、筆者らが所属する研究室のゼミ4回で使用した。メンバーは1名の教員（本稿第2著者）、5名の博士課程学生（うち1名は本稿第1著者）、修士課程学生11名で行われた。修士課程学生のうち3名が女性（うち2名は中国語が母語）で、それ以外は全員男性（うち1名は中国語が母語）である。実験期間中に欠席や見学者によって若干の参加者変動がある。

進捗報告ゼミでは、1回のゼミあたり研究室のメンバー4～6名が進捗報告を行う。進捗報告は意見収集型プレゼン形式で実施される。発表予定者は1人ずつ順に登壇し、まず現在取り組んでいる作業の内容と進捗状況について、前回の進捗報告からの差分だけではなく、外部発表と同様に背景、目的などを含み研究全体が把握できるように発表する。その後、聴衆を交えて口頭での質疑応答と意見交換を行う。Chatplexerは、この発表と質疑応答の間を通じて使用される。

発表者の観点に基づく重要な発言を得るために、ゼミ終了後に発表を行った発表者にチャット・ログを印刷した物を渡し、意見や質問として自分自身の研究にとって重要であると思われるチャット発言をピックアップしてもらった。ただし、重要さを段階的に重み付けすることは求めず、重要か否かだけの判断を求めた。このピックアップ作業は十分に時間的余裕を与えて行った。

Chatplexerのログは、人間が投稿したチャット発言と、発表者用アプリケーションが自動的に投稿したスライド発言が混在した状態のものとなるが、以降の分析は全て、人間が投稿したチャット発言のみを対象としたものである。

5.2 収集したデータの概要

4回の進捗ゼミで、15人の発表者によって計20回の発表が行われた。全部で2496発言のチャット・ログと、248発言の重要発言が得られた。発言種別毎の内訳はクロスチャンネル返信が510発言(20.4%)、新規発言が312発言(12.5%)、通常の返信が1674発言(67.1%)であった。括弧内は内訳である。

平均すると、発表者1人あたり10分17秒の発表を行いその後27分1秒の質疑応答をしており、全体で37分18秒であった。その約37分間に平均124.8発言のチャット発言が送信された。1分あたりの平均発言数は、発表中は3.4発言/分、質疑中は3.2発言/分、全体では3.3発言/分であった。チャット発言全体のうち重要発言が占める割合(重要発言占有率)は、発表中が9.3%、質疑中が10.6%、全体では9.9%であった。

チャット発言の返信関係は、メールのスレッドと同様に木構造を為している(図7)。クロスチャンネル返信発言をルートとし、これに対する通常返信で構成される構造木(クロスチャンネル返信木)と、新規発言をルートとして、これに対する通常返信で構成される構造木(新規木)を考える。意味的に見れば、クロスチャンネル返信木はフロントチャンネルである口頭対話あるいはスライドの内容に対する返信及び意味的に連鎖する発言によって構成され、新規木はフロントチャンネルとは無関係な内容の発言及び意味的に連鎖する発言によって構成されている。

```
[542] zawareo: 重要発言数ってメモボチられた発言? >>>*
[544] 田中: これ言っているのか解らないですけど この前のアンケートで >>>542
[546] zawareo: なるほろ >>>544
[545] ふじた: 発表者による評価ではないでしょうか 終わった後のアンケートの >>>542
[548] ky: 重要発言は本当に重要なのか >>>*
[553] ky: まあ、発表者から見た場合だから正しいのかー >>>548
[554] Kazz: だと >>>553
[557] 田中: 発表者が質疑応答時に取り上げたい発言 = 重要発言だったかと >>>553
[558] ちば: まったくなくなったら悲しいな... ときどき >>>557
```

図7 返信による木構造の例

クロスチャンネル返信と新規発言がそれぞれの構造木のルート・ノードとなるので、クロスチャンネル返信木は510本(62.0%)、新規木は312本(38.0%)である。発言単位では、1503発言(60.2%)がクロスチャンネル返信木に属しており、993発言(39.8%)が新規木に属している。発表者1人あたりでは平均で54.8本の返信木があった。クロスチャンネル返信木の数、ならびにクロスチャンネル返信木に属する発言の数は、いずれも過半数を占めており、チャット上ではフロントチャンネルである口頭対話の内容に関連する発言が多くやりとりされていることが分かる。

クロスチャンネル返信木に含まれる重要発言と新規木に含まれる重要発言を比較すると、211発言(85.1%)の重要発言がクロスチャンネル返信木に属しており、新規木に属する重要発言は37発言(14.9%)であった。重要発言占有率で見ると、クロスチャンネル返信木の重要発言占有率が平均14.0%であるのに対し、新規木の重要発言占有率は3.7%であった。このように、重要な発言は新規木にはあまり存在せずクロスチャンネル返信木に多く存在しているが、クロスチャンネル返信木の数自体が多いため、クロスチャンネル返信木における重要発言占有率が大きく高いというわけではない。

6. 重要発言推定実験

実験データに含まれる返信木の情報を元に、重要な発言を推定できるかどうかを検証した。推論アルゴリズムとしては、決定木を用いる。モデルの作成と検定にはWekaを使用し、決定木はJ4.8アルゴリズムを使用して作成する。J4.8はJavaによるC4.5の実装である。

本論文では、文法構造や言葉の意味には立ち入らず、主として返信の形態に基づいた重要発言の推定を試みる。クロスチャンネル返信を含め、返信の形態が言語によって異なるとは考えにくい。ゆえに本手法によって、言語種別に依存しない重要発言推定が可能となると考えられる。そこで、返信形態以外のパラメーターとしては、やはり言語種別に依存しない、発言の文字数とURLを含むかどうかというパラメーターのみを採用した。

検定は、取得したデータを10分割してそれぞれを1回ずつテストデータとする10-fold

交差検定により行った。クロスチャンネル返信に関するパラメーターを含む以下の7つのパラメーターを学習に用いた場合と、クロスチャンネル返信に関するパラメーターを含まない6つのパラメーターを用いた場合とで、最終的に重要な発言として選択されたかどうかを推定させた場合の精度を比較する。

- 言語に依存しない基本のパラメーター
 - － 発言の文字数
 - － 発言が URL を含むかどうか
- 返信関係の構造木に関するパラメーター
 - － 発言の構造木における階層深さ
 - － 発言に対する返信（子ノード）の数
 - － 発言が属する構造木が全体でどの程度の葉ノードを持っているか
- クロスチャンネル返信に関するパラメーター
 - － **クロスチャンネル返信木であるかどうか**
- その他のパラメーター
 - － 発言の入力にかかった時間

J4.8 による決定木では、クロスチャンネル返信木であるかどうかのパラメーターを用いる場合、適合率 61.5%，再現率 12.9%，F 値 0.213 であった。この時の決定木を図 8 に示す。図中の括弧は条件に合致したデータの数を表し、誤って分類されたデータがあれば「/」の後ろにそのデータ数が表される。一方、クロスチャンネル返信木であるかどうかのパラメーターを用いない場合は、適合率 28.6%，再現率 1.6%，F 値 0.031 であった。

7. 考 察

J4.8 ではクロスチャンネル返信木であるかどうかという、たった 1 つのパラメーターを取り除いただけで、適合率が 61.5% から 28.6% へと半分以下になってしまった。ここからクロスチャンネル返信木であるかどうかという情報が、適合率に大きな影響を与えることが分かる。

しかし再現率はクロスチャンネル返信木かどうかの情報を用いた場合でもかなり低い結果となった。使用しているデータでは、発表 1 回あたり平均して 124.8 発言のチャット発言があり重要発言占有率が 9.9% であることから、チャット・ログには平均して 12.4 発言の真の重要発言が含まれていると推測される。クロスチャンネル返信木であるかどうかのパラメーターを用いた場合は、再現率が 12.9%，適合率が 61.5% であるので、2.6 発言の重要発言を

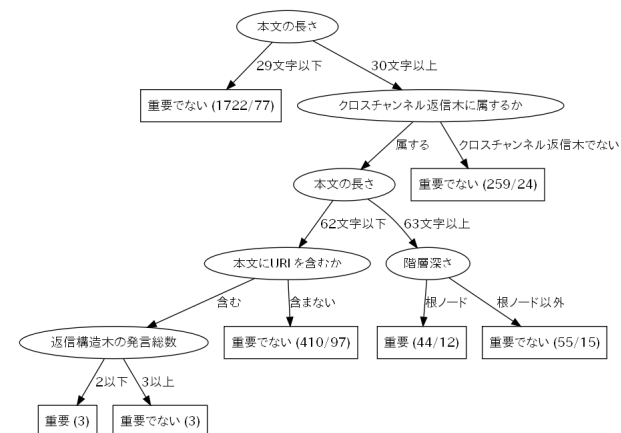


図 8 決定木 クロスチャンネル返信情報あり

推定し、うち 1.6 発言が真の重要発言である。

一般に時間的制約が大きい学会発表などでの使用を想定すると、既存の口頭による意見・質疑に加え、さらにチャットからも意見・質疑を得るということになる。筆者らの経験では、本論文が想定する学会発表のようなプレゼン型会議においては、チャットから 3 発言程度の重要発言を推定するだけでも必要十分なケースは多いと思われる。ただし具体的な数については、口頭発言による質疑意見を差し置いてチャット発言を取り上げなければならないほど重要なのかといった複雑な問題を含んでくるため、今後も検討を加える必要がある。また、発表後に重要な発言を参照するという使用形態を想定した場合は、もっと再現率を高める必要があるだろう。

8. ま と め

本論文が提案するクロスチャンネル返信は、チャット上（バックチャンネル）から対面口頭対話上（フロントチャンネル）の内容に対しての明示的な返信を可能にする。クロスチャンネル返信機能を実装した Chatplexer を制作し、研究室のゼミで 2ヶ月にわたって使用した。このデータからは、以下のことが分かった。

- クロスチャンネル返信木の数、およびクロスチャンネル返信木に属する発言の数は、共に過半数を占めている。

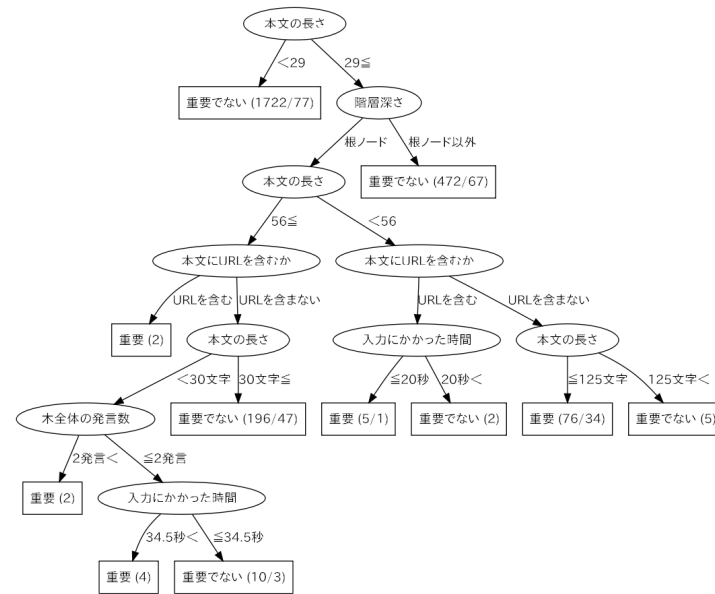


図9 決定木 クロスチャンネル返信情報なし

参考文献

- 1) Rekimoto, J., Ayatsuka, Y., Uoi, H. and Arai, T.: Adding another communication channel to reality: an experience with a chat-augmented conference, *CHI '98: CHI 98 conference summary on Human factors in computing systems*, New York, NY, USA, ACM, pp.271–272 (1998).
- 2) 綾塚祐二, 河口信夫: 参加者が作る会議支援システム〜WISS Challenge〜, コンピュータソフトウェア, Vol.23, No.4, pp.76–81 (2006).
- 3) Golub, E.: On audience activities during presentations, *J. Comput. Small Coll.*, Vol.20, No.3, pp.38–46 (2005).
- 4) 平光節子, 白井正博, 杉山岳弘: チャットをベースにした会議のコミュニケーション活性化システムの検討, 情報処理学会研究報告. HI, ヒューマンインタフェース研究会報告, Vol.2003, No.94, pp.7–12 (20030926).
- 5) Hembrooke, H. and Gay, G.: The Laptop and the Lecture: The effects of multi-tasking in learning environments, Vol.15 (2003).
- 6) 百合山まどか, 畠中晃弘, 垂水浩幸, 上林彌彦: チャットを利用した学生間コミュニケーション促進の実験, 情報処理学会研究報告. [グループウェア], Vol.2000, No.97, pp.37–42 (20001019).
- 7) Harry, D., Green, J. and Donath, J.: backchan.nl: integrating backchannels in physical space, *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, New York, NY, USA, ACM, pp.1361–1370 (2009).
- 8) 西田健志, 栗原一貴, 後藤真孝: On-Air Forum: リアルタイムコンテンツ視聴中のコミュニケーション支援システム, WISS2009 第17回インタラクティブシステムとソフトウェアに関するワークショップ論文集, pp.59–100 (2009).
- 9) 田中充勅使河原可海平島大志郎: 会議録としての連続メディア情報の重要度を用いた検索方式の比較検討, 分散協調とモバイル (DICOMO2003) 予稿集, pp.353 – 356 (2003).
- 10) 由井蘭隆也, 重信智宏, 榎野晶文, 宗森 純: リアルタイムなコミュニケーション行為であるチャットへの意味タグ付加と電子ゼミナールへの適用, 情報処理学会論文誌, Vol.47, pp.161 – 171 (2006).
- 11) McCarthy, J.F. and boyd, d.m.: Digital backchannels in shared physical spaces: experiences at an academic conference, *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, New York, NY, USA, ACM, pp.1641–1644 (2005).

- 発表者にとって重要な発言は新規木にはあまり存在せずクロスチャンネル返信木に多く存在している。

また、クロスチャンネル返信木の情報を用いると重要発言推定にどのような影響を与えるかについて検証したところ、クロスチャンネル返信木であるかという情報を用いると重要発言推定における適合率が大きく改善するということが分かった。更に複数の言語的パラメータと組み合わせることにより、実用的なレベルで重要発言推定システムを作ることが可能であると考えられる。

今後は日本語環境に限定し、重要発言がどのような言語的特徴を持っているかを検討することで重要発言推定システムを実装し、発表者とシステムが相互に関わり合う中でより良いチャット併用会議ができるかどうかを検討したい。