平成21年度 修士論文

# チャットルームにおけるユーザの特徴を用いたトピック抽出システム

電気通信大学 大学院情報システム学研究科

情報システム基盤学専攻

0853007 川端　聖

指導教員　多田 好克　教授
村山 隆彦　准教授
古賀 久志　准教授

提出日　平成22年1月28日

# 目 次

# 図 目 次

# 第 1 章

# はじめに

インターネットの普及に伴い、インターネットを利用したコミュニケーションが増大している。これは、電子メールのような 1 対 1 で行うものもあれば、電子掲示板のような、主に不特定多数の中で行うものもある。特に近年、Blog、SNS、Twitter[1]、チャット等、様々な形態の不特定多数のコミュニケーションを含むサービスが注目を浴びている。本研究では、リアルタイムに行うコミュニケーションへの興味の高まりから、この内のチャットに着目する。

チャットとは、ネットワーク上に用意された 1 ヵ所のスペースにおいて、複数のユーザが、リアルタイムにテキストベースの会話を行うシステムのことである。チャットを行うユーザは、チャットルームと呼ばれる場所で会話を行う。チャットルームには、テーマが静的に付けられている。テーマとは、そのチャットルームで推奨される会話の内容をユーザに伝えるものである。たとえば、食べ物、政治、経済のようなテーマがあり、ユーザはこれらのテーマから自分の好みのチャットルームを選択する。

不特定多数の人が集う大規模なチャットには、各ユーザの興味を満たすために様々なテーマのチャットルームが存在する。しかし、実際のチャットルームでは、そのチャットルームのテーマに沿った、自分の興味のある話をしているとは限らない。逆に、自分の興味のないテーマのチャットルームで、自分の興味のある話をしていることもある。また、最初は興味のある話をしていても、人の入れ替りや時間の経過とともにチャットルームの話題が変わる可能性があるため、興味のない話に

変わってしまうことがある。つまり、テーマによってチャットルームを選んでも、自分に最も適しているチャットルームに行けるわけではないという問題がある。

チャットルームのテーマと実際の会話の内容が違う例を、図を用いて説明する。図 1.1 において、政治のテーマのチャットルーム 1 では、政治と関係のないペットの話がされている。一方、政治と関係のない食べ物のテーマのチャットルーム 2 では、政治の話がされているということもある。これが、チャットルームのテーマと実際の会話の内容が違うということである。このような状態であれば、政治の話を期待して、政治のチャットルームに入室したユーザ A は、政治とは関係のないペットの話しか聞けない。また、他のチャットルームで、何の会話が行われているかもわからないので、食べ物のチャットルームでユーザ A の求める政治の話を聞ける機会も逃すこととなる。

そこで本研究では、チャットルームの特徴やチャットルームに参加している各ユーザに着目し、この問題の解決に取り組む。

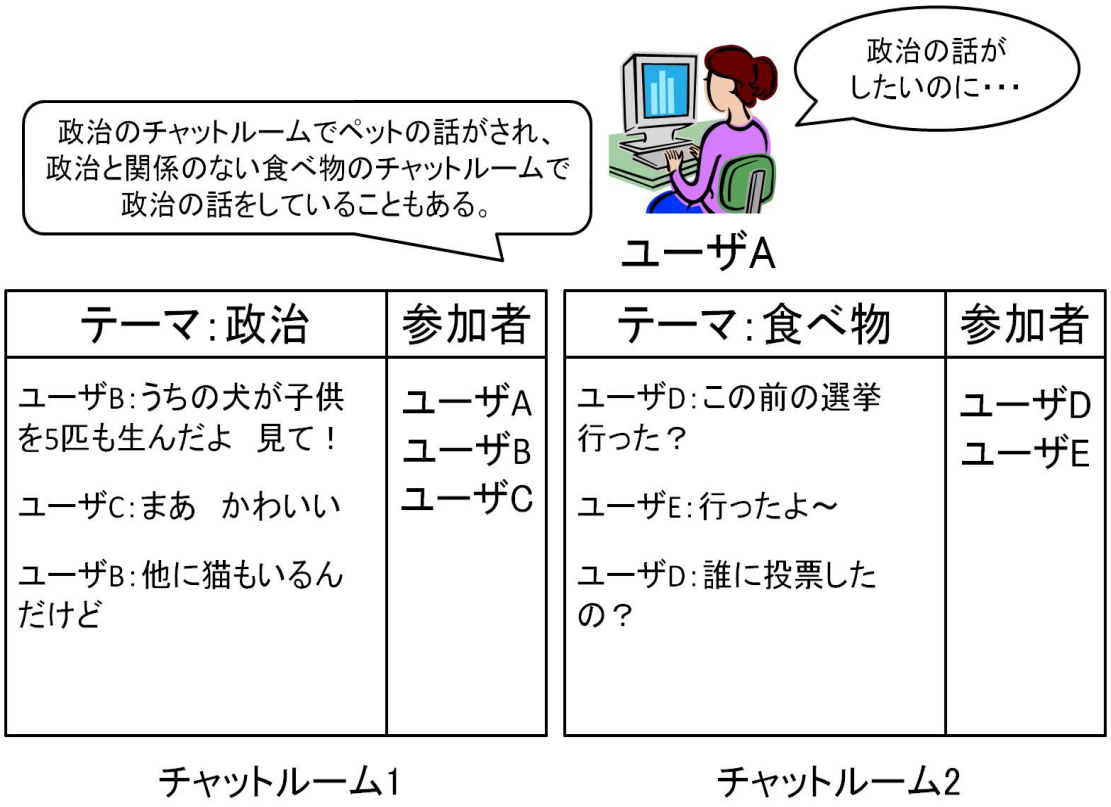


図 1.1: チャットルームのテーマと会話の内容が違う例

本論文は、次のような構成とする。第 2 章では、研究概要を説明し、チャットの問題、チャットの問題から導かれる課題、課題に対するアプローチを議論する。第 3 章では、関連研究を説明し、本研究と比較する。第 4 章では、システム設計について議論する。第 5 章では、実装について説明する。第 6 章では、実装したシステムを用いた本システムの有効性を測るための実験について説明する。第 7 章では、実験の結果とその評価について議論する。第 8 章では、実装したシステムの問題点を挙げ、議論する。最後に第 9 章で、まとめを議論する。

# 第 2 章

# 研究概要

本章ではまず、研究の概要を説明する。次に、チャットの特徴、チャットの特徴から導かれる課題、課題に対するアプローチを議論する。

第 1 章で説明したように、テーマによってチャットルームを選んでも、自分に最も適しているチャットルームに行けるわけではないという問題がある。この問題から、各チャットルームで実際に話されている内容にふさわしいテーマを知り、自分に合ったチャットルームを選びたいという要求がある。そこで本研究では、会話の内容から動的に決まるトピックをテーマとして代用し、チャットルームのトピックを抽出することで、各チャットルームで実際に話されている内容にふさわしいテーマを見つけることを目的とする。

なお、ここで言うトピックとは、チャット内での会話を特徴付ける名詞群の上位概念のカテゴリを指す。たとえば、「みかん」「りんご」「ラーメン」等の名詞群は、「食べ物」という上位概念のカテゴリに属し、「選挙」「法案」「国会」等の名詞群は、「政治」という上位概念のカテゴリに属する。この例での、「食べ物」や「政治」がトピックであり、それをテーマの代用とする。

## 2.1 チャットの特徴

チャットのログ (発言そのものの記録) には、チャット特有の性質がある。トピックを抽出する際に、主に以下の 2 つの特徴に着目し、研究の課題を導いた。

**特徴**1　話題が短時間で変わってしまうことがある。
チャットでは、話題が変わってしまう機会がよくある。たとえば、会話を行っているユーザが、現在の話題に興味を失い、別の話題を提示することがある。またチャットルームには、不特定多数の人が、短期間で出入りするため、人の入れ替わりによっても、話題が変わることがある。このような特徴から、チャットでは、トピックの抽出を行うことのできる期間が、短期間になってしまいやすい。

**特徴**2　文として整っていないものが多いので、特徴を捉えることが難しい。
電子メールでは、相手の宛先や件名、本文を書いてから、相手に自分の発言を伝えなければ成らない。電子掲示板でも、自分の名前や件名、本文を書いてから、自分の発言を書かなければならない。これらの媒体は、自分の発言を行うまでに、一定の敷居がある。

一方、チャットでは、本文を書くだけで良く、対話形式で、自分の発言を修正できたり、相手とのメッセージのやりとりがリアルタイムでできたりするので、電子メールや電子掲示板よりも、ユーザが気軽に発言する傾向にある。よって、くだけた表現を含むこともあり、文として整っていないものも多い。このような文は、チャットルームに参加していないユーザにとっては、理解できないことがある。また、自然言語の解析器で解析し難いため、文を計算機で処理し難い。このような特徴から、チャットでは、トピックの抽出として、文のすべての要素を考慮するのは、困難だとと考えられる。

## 2.2　チャットでトピックを抽出する際の課題

チャットの特徴を踏まえると、トピックを抽出する際の課題は主に以下の2つである。

**課題**1　短期間の少ない情報量のログで会話の特徴を抽出しなければならない。
チャットでは話題が短期間で変わってしまうことがよくあるため、話題となるログの量が少ないことが多い。この少ない情報量のログの中から、トピックとなる会話の特徴を抽出しなければならない。

**課題**2　文の特徴を捉えることが難しいものでも、うまく特徴を捉えなければならない。
チャットではくだけた表現を使うこともあり、文として整っていないものもある。このようなものは自然言語の解析器で解析し難いため、文のテキストを計算機で処理し難い。しかし、チャットでトピックを抽出するためには、このような文の特徴もうまく捉えなければならない。

## 2.3　アプローチ

短期間の少ない情報量で、特徴が捉え難いチャットのログからトピックを抽出するには、まず、少ない情報量からより多くの特徴を抽出するために、過去の各ユーザの特徴と過去のチャットルームのログの特徴に着目する。そして、特徴が捉え難いチャットのログの特徴をうまく捉えるために、現在のチャットルームのログの特徴に着目する。そこで本研究では、各ユーザの特徴とログの特徴を抽出し、それらをマージすることで、チャットでトピックを抽出する課題を解決する。

**各ユーザの特徴抽出**　少ない情報量でのトピック抽出の課題に対するアプローチとして、各ユーザの特徴である過去の発言履歴を抽出する。本研究では、チャットルームの会話に参加している各ユーザの特徴を抽出することで、会話に参加している各ユーザのログの意味を拡大する。

たとえば、あるユーザが、現在、「裏金」という名詞だけを含む発言しかしていなかったとする。「裏金」とは幅広い分野に使われる名詞である。つまり現在の名

詞だけでは、どのようなトピックの発言を行っているか定かではない。しかし、そのユーザが、過去の発言履歴に「国会」や「与党」などという「政治」に関わる名詞を頻繁に使っていたとしたら、現在の「裏金」という名詞も政治に関わる発言の可能性が高いと判断できる。これが、各ユーザのログの意味を拡大するということである。

これによって、現在のチャットルームのログの行数が少なくても、より多くの会話の特徴を抽出することができると考える。

**ログの特徴抽出**　少ない情報量でのトピック抽出、文の特徴が捉え難いもののトピック抽出の課題に対するアプローチとして、チャットルームのログの特徴を抽出する。

これによって、少ない情報量でのトピック抽出の課題に対して、現在のログの意味を拡大し、短期間の少ない情報量のログで会話の特徴を抽出する。また、文の特徴が捉え難いもののトピック抽出の課題に対しては、比較的抽出が容易な名詞だけに着目することで対処する。

# 第 3 章

# 関連研究

本章では、関連研究を説明し、本研究との違いを述べる。

野美山 [2] は、キーワードの問合せに対して、そのキーワードについてのトピック抽出を行っている。ここではまず、キーワードの頻度が低い基準である恒常的出現頻度を推定する。そして、恒常的出現頻度に対して、最も大きな頻度を持つ時刻をトピックの開始時刻とする。その後、キーワードの頻度が、単調減少を終えるか恒常的出現頻度を下回った時をトピックの終了時刻として、トピックを抽出している。

トピック抽出の基準として、恒常的出現頻度を推定していることに対して、本研究では、一定時間内のチャットルームの発言数や抽出された名詞数に閾値を設けることで、ログがトピックとして成り立つ基準を設けている。これによって、会話として成り立っていない、多くの名詞を含む一つの長い発言だけでも、トピックと判断することを回避できる。

関口ら [3] は、blog 発信者の特徴を利用してトピック抽出を行っている。ここでは blog 発信者の興味を抽出し、blog 発信者間の興味の類似性によって、blog 記事の中に現れる語句に重み付けを行っている。つまり、興味の類似性の高い発信者間で共通して使われる語句は、より話題性がある語句だと認識される。このような重み付けを行うことで、ブログを発信している人数が少ないマイナーな分野も含めた様々な分野のトピックを抽出している。

blog 発信者と他の発信者の特徴を用いたトピック抽出に対して、本研究でも、

チャットルームで会話を行っている各ユーザの過去のログから各ユーザの特徴を用いたトピック抽出を行う。

石井ら [4] は、電子掲示板の書込みの盛り上がりを考慮したトピック抽出を行っている。ここでは、電子掲示板に書き込まれた記事のキーワードが、後続の書込みによってどれだけ引用されているかによって、書き込まれた記事の盛り上がりを測っている。このような盛り上がりを考慮したトピック抽出を行うことで、ユーザの飽きがこないトピックの提示を実現している。

この研究では、盛り上がりを測る際に後続の書込みを利用している。つまり、未来の書込みを考慮しなければならない。しかし、本研究の対象であるチャットは、電子掲示板とは違い、現在のトピックを推定しなければならないため、未来のログを考慮するのは適当ではない。そのため、現在のログと過去のログだけを用いて、トピック抽出を行う。

野美山 [2]、関口ら [3]、石井ら [4] は、それぞれ、新聞記事、blog、電子掲示板のトピック抽出を行っている。これらの媒体は、チャットに比べ、一続きのトピックに関する情報が多かったり、文として整っていたりする。本研究では、各ユーザの特徴とチャットルームのログの特徴に着目することで、これらに比べ、より少ない情報量でより特徴が捉え難い媒体であるチャットのトピックを抽出する。

表 3.1 と表 3.2 はそれぞれ、実際のチャットログとニュース記事のすべての品詞に対しての記号と格助詞の構成を表している。この図から、トピックが抽出し易いとされるニュース記事と比較して、実際のチャットログの方が、格助詞が少ない等、文の品詞構成が特徴的である。また、記号の比率が高いので、特徴が捉え難いことがわかる。また、付録 A の図 A.1 と図 A.2 では、実際のチャットログとニュース記事のすべての品詞構成を表している。

表 3.1: チャットログのすべての品詞に対する記号と格助詞の構成 (%)

| 品詞 | 構成比 |
|---|---|
| 記号–文字 | 15.49 |
| 記号–一般 | 0.29 |
| 助詞–格助詞 | 6.09 |

表 3.2: ニュース記事のすべての品詞に対する記号と格助詞の構成 (%)

| 品詞 | 構成比 |
|---|---|
| 記号–文字 | 0.15 |
| 記号–一般 | 0.02 |
| 助詞–格助詞 | 17.02 |

# 第 4 章

# システム設計

本章では、アプローチを実現するためのシステム設計について議論する。

## 4.1 システム要件

第 2 章で述べたように、本研究では、各ユーザの特徴とログの特徴を抽出し、それらをマージするという手法を使う。ここでは、各ユーザの特徴とログの特徴を抽出するためのシステム要件を示す。

### 4.1.1 各ユーザの特徴抽出

各ユーザの特徴抽出を行うために、以下の要件について議論する。

**要件 1** 過去の発言履歴から各ユーザを特徴付け、そのログを現在の会話を構成する一部として扱う。

**要件 1 に対する議論** これは、少ない情報量でのトピック抽出の課題の解決方法である。現在のチャットルームで会話を行っている各ユーザの過去の発言を、現在の各ユーザの発言の一部とする。これによって短期間の少ない情報量のログを拡大する。この方法の欠点は、抽出されたトピックが各ユーザの過去の発言によって引きずられてしまう可能性があることである。トピックが引きずられるとは、各ユーザの過去の発言によるトピックによって、現在のチャットルームに適切だと思われる

トピックが、変わってしまう可能性があるということである。この欠点に対して、抽出されたトピックが引きずられてしまっても、各ユーザの過去の発言によって生じた過去のトピックは、現在のチャットルームに反映されている。よって、その過去のトピックが、話し易くなっているため、そのトピックを話す土壌は整っていると考えられる。よって、この欠点を補うことができる。

### 4.1.2 ログの特徴抽出

ログの特徴抽出を行うために過去のログと現在のログの両方に着目することで、より特徴を捉えることができると考えられるため、2つの要件がある。以下、2つの要件について議論する。

**要件 2** 現在のチャットルームの直前のログを、現在の会話を構成する一部として扱う。

**要件 2 に対する議論** これは少ない情報量でのトピック抽出の課題の解決方法である。現在のチャットルームの直前のトピックは、現在も話されている可能性が高い。よって、現在のチャットルームの直前のログを現在のチャットルームの会話の一部とする。これによって、短期間の少ない情報量のログを拡大する。この方法の欠点も、抽出されたトピックが直前の会話によって引きずられてしまう可能性があることである。この欠点に対して、抽出されたトピックが引きずられてしまっても、直前の会話を取り入れることで、より継続性のあるトピックを抽出できると考えられる。よって、この欠点を補うことができる。

**要件 3** 他の品詞に比べ、比較的抽出が容易な名詞だけに着目して、トピックとなるいくつかの名詞群に分類する。

**要件 3 に対する議論** これは、文の特徴を捉えることが困難なものの特徴を捉える課題に対する解決方法である。特徴が捉え難いチャットのログでも、抽出が容易な

名詞だけを抽出し、その名詞をトピックとなる名詞群に分類することで、会話の特徴を捉えることができる。この方法の欠点は、名詞だけ抽出することを考えることで他の品詞や句の特徴を無視してしまうことである。しかし、特徴が捉え難い文が多いため、名詞以外のものも用いるとノイズの影響によりうまくトピック抽出を行えない可能性がある。よって、特徴が捉え難い文が多いチャットのような媒体のトピック抽出では、ノイズの影響のできるだけ少ない方法を選択することが妥当だと考える。

## 4.2 システム概要

前節で述べた3つのシステム要件から、以下の2種類の入力と5つの処理が必要となる。システムの概要は、図4.1に示す。図4.1における入力及び処理は以下の通りである。

**入力1** 過去の発言履歴データ

**入力2** 現在のトピック推定用データ

**処理1** 各ユーザが過去に発言した名詞の抽出

**処理2** 現在の会話の直前に行われた会話の名詞の抽出

**処理3** 現在のチャットルームで行われた会話の名詞の抽出

**処理4** 各抽出データのマッチング

**処理5** 現在のチャットルームのトピックの推定

処理1では、各ユーザの特徴抽出を行い、処理2、処理3では、ログの特徴抽出を行う。これらの処理はトピック抽出の前処理であり、各ユーザの特徴とログの特徴を現在のチャットルームのログにマッチングし易くする。そして、処理4では、

処理1、処理2、処理3の出力データをマージする。この処理によって、各ユーザの特徴とログの特徴を、現在のチャットルームのログに取り入れたデータができる。最後に処理5で、マージした結果から現在のチャットルームのトピックを推定する。このような流れで処理を行うことで、各ユーザの特徴とログの特徴によって少ない情報量のログの意味を拡大する。特徴の捉え難いものの特徴も取り入れられるので、チャットに対して最適なトピックを抽出できる。

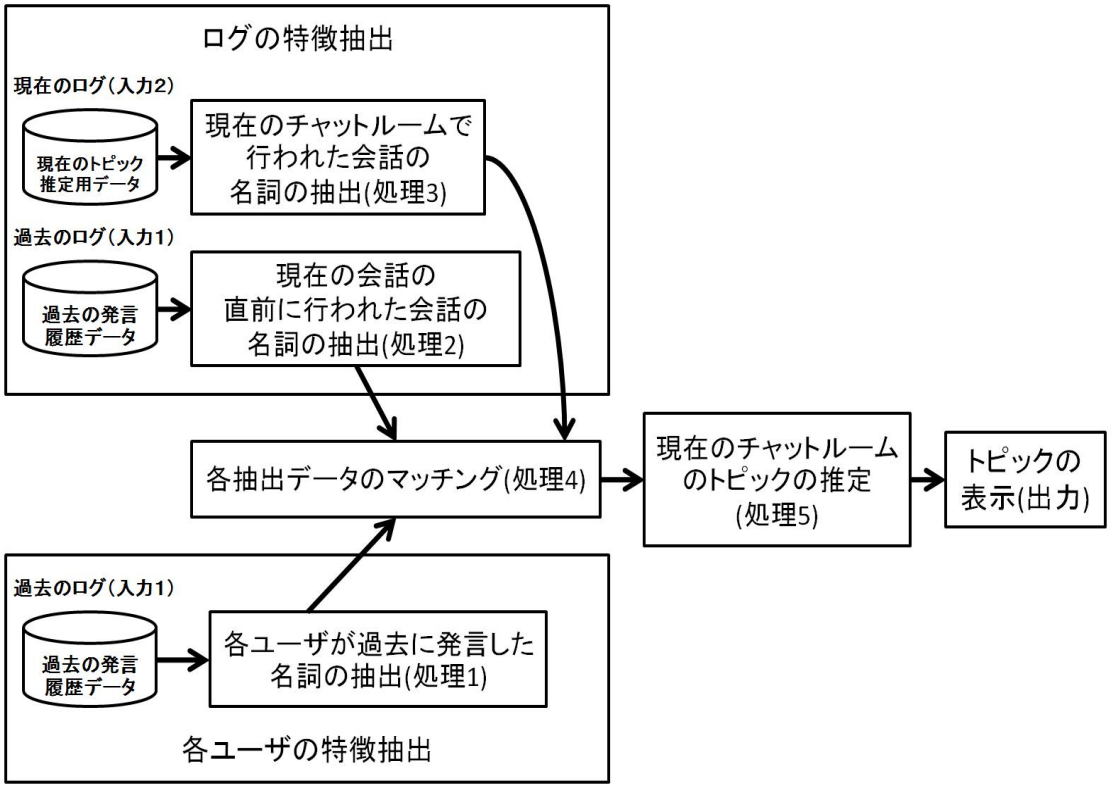


図 4.1: システムの概要図

## 4.3 システムの各処理

図 4.1 の 2 種類の入力と 5 つの処理について、それぞれ詳しく説明する。

**入力** 1　過去の発言履歴データ

過去の発言履歴データは、各ユーザが過去に発言した名詞の抽出、現在の会話の直前に行われた会話の名詞の抽出の処理の入力に用いられる。各ユーザが過去に発言した名詞の抽出の処理の入力では、現在、会話を行っている各ユーザの特徴として、各ユーザの過去の名詞を抽出するために用いる。現在の会話の直前に行われた会話の名詞の抽出の処理の入力では、現在のチャットルームのログの特徴として、現在の会話の直前に行われた会話の名詞の抽出に用いる。このデータによって、現在のチャットルームのログの意味を拡大する。

**入力** 2　現在のトピック推定用データ

現在のトピック推定用データは、現在のチャットルームで行われた会話の名詞の抽出処理の入力に用いられる。これは、現在のチャットルームのログの特徴として、現在のチャットルームで行われた会話の名詞を抽出する処理に用いられる。現在のトピック推定用データの内、既に現在のチャットルームのトピックの推定に用いられたデータは過去のデータとなるため、以後、過去の発言履歴データとして用いる。このデータによって、現在のチャットルームのトピックを推定する。

### 4.3.1 各ユーザが過去に発言した名詞の抽出

この処理は、各ユーザごとの発言履歴の名詞を抽出し、この先の処理で現在のチャットルームのログの一部としているため、要件 1 と要件 3 を満たし、図 4.2 のように処理される。この流れ図ではまず、過去の発言履歴データをユーザごとのデータに分割する。

次に、この分割したデータの各々に形態素解析を行い、各ユーザの特徴である名詞を抽出する。

この処理はトピック抽出の前処理であり、各ユーザの特徴を現在のチャットルームのログにマッチングし易くする。

**形態素解析**　形態素解析とは、自然言語で表された文を意味の最小の単位である形態素に分割し、各形態素に品詞を付与することである。この解析を手作業で行うと膨大な時間と手間がかかってしまう。そのため本研究では、形態素解析器を使ってこの作業を行う。しかし、自然言語の本質は曖昧性にあり、形態素解析においても最適な解を見つけることが困難な場合があり、形態素解析器の結果は完全なものではない [5]。

### 4.3.2　現在の会話の直前に行われた会話の名詞の抽出

この処理は、現在の直前のログの名詞を抽出し、この先の処理で現在のチャットルームのログの一部としてているため、要件 2 と要件 3 を満たし、図 4.3 のように処理される。

この流れ図ではまず、ログを時系列順に処理するために現在のトピック推定用データを一定の時間間隔に分割する。本研究の実装ではこの時間間隔を 5 分とした。このように、短期間だと考えられる 5 分の時間間隔に分割することよって、チャット特有の特徴である短期間でトピックが変わりやすい環境に対応する。

次に、現在の会話の直前のログの名詞を抽出するために、一番最新以外の各分割ログに対して形態素解析を行い、ログの特徴である名詞を抽出する。

この処理はトピック抽出の前処理であり、過去のログの特徴を現在のチャットルームのログにマッチングし易くする。

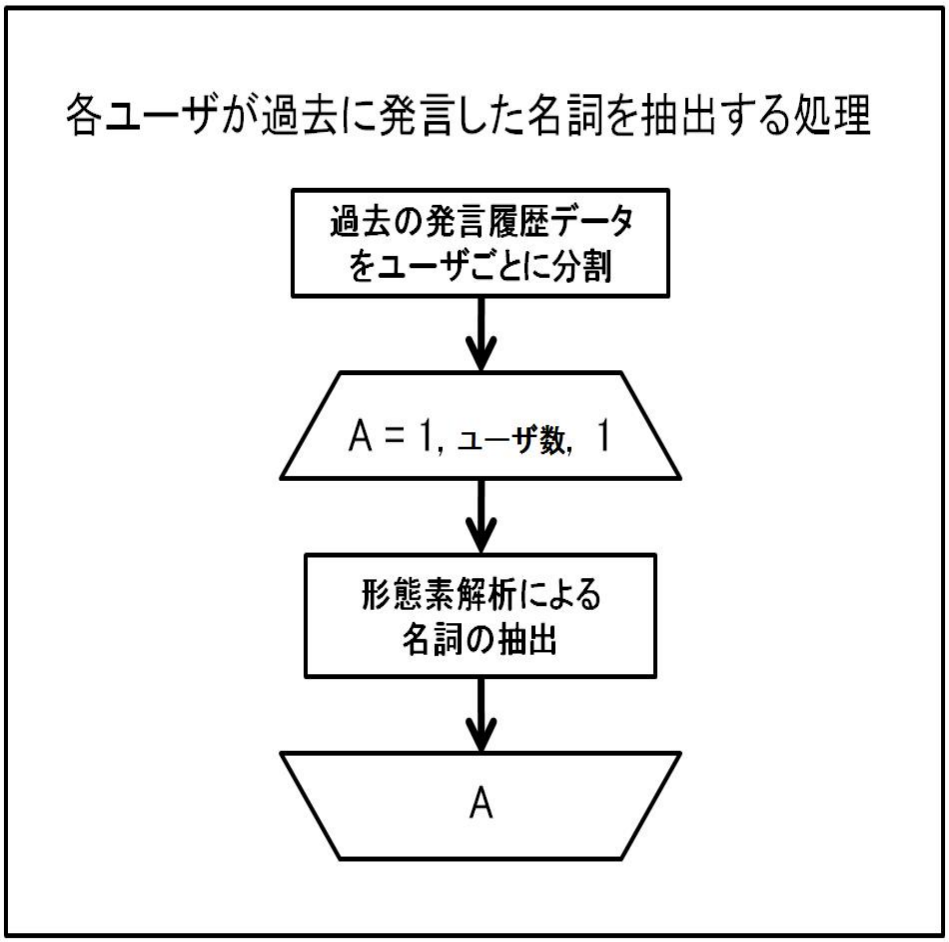


図 4.2: 各ユーザが過去に発言した名詞の抽出処理

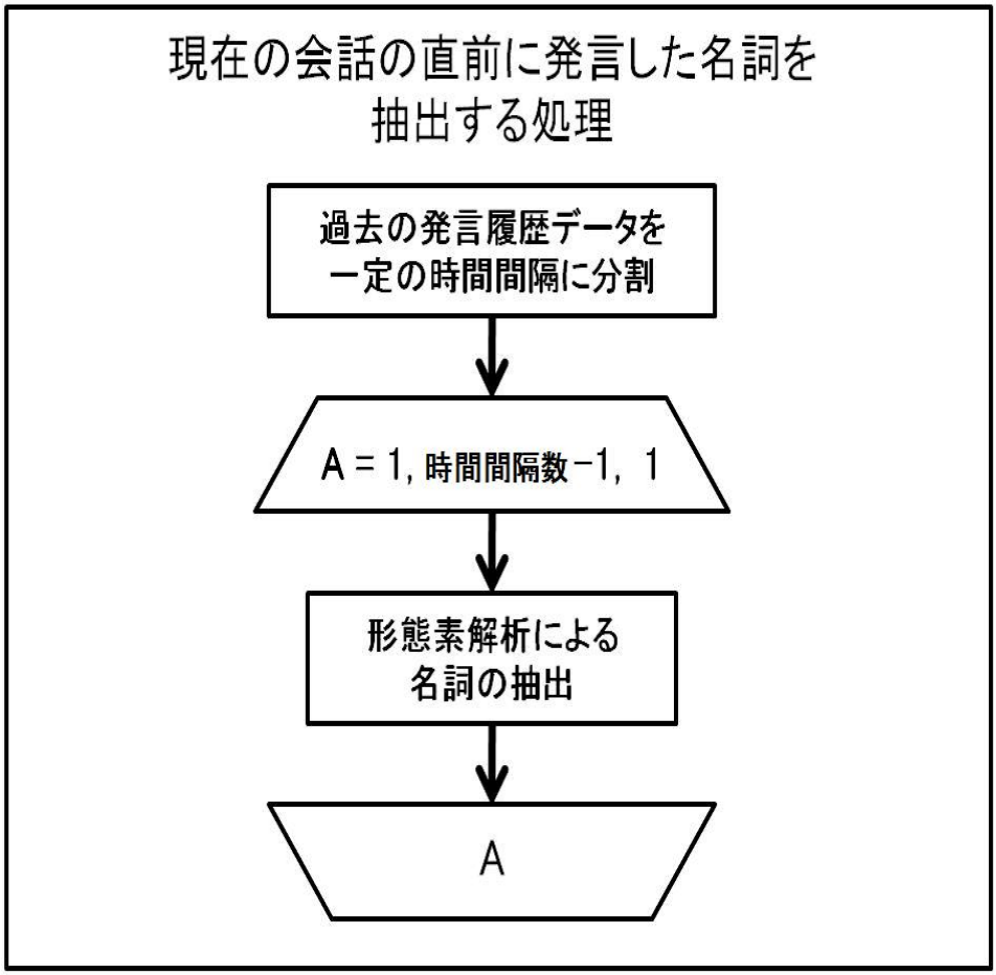


図 4.3: 現在の会話の直前に行われた会話の名詞の抽出処理

### 4.3.3 現在のチャットルームで行われた会話の名詞の抽出

この処理は名詞の抽出を行っているため、要件 3 を満たし、図 4.4 のように処理される。この流れ図ではまず、ログを時系列順に処理するために、現在のトピック推定用データを一定の時間間隔に分割する。これにより、短期間でトピックが変わりやすい環境に対応する。次に、現在のチャットルームの名詞を抽出するために、各分割ログに対して形態素解析を行い、ログの特徴である名詞を抽出する。そしてこの後の処理で、各ユーザが過去に発言した名詞の抽出と現在の会話の直前に行われた会話の名詞の抽出処理の出力データをマッチングするために、行項目に形態素解析によって抽出した名詞、列項目に分割した各時間とり、行列化したデータを作る。

たとえば、行列化したデータは表 4.1 のようになる。表の各要素は、ある時間のログに存在したある名詞の個数を指す。行列化したデータは疎行列になる傾向がある。この処理はトピック抽出の前処理であり、現在のチャットルームのトピック推定を行い易くする。

表 4.1: 行列化したデータの例

| | 時間 1 | 時間 2 | 時間 3 | … | 時間 n |
|---|---|---|---|---|---|
| 名詞 1 | 2 | 0 | 0 | … | 0 |
| 名詞 2 | 0 | 0 | 0 | … | 1 |
| 名詞 3 | 0 | 0 | 0 | … | 0 |
| … | … | … | … | … | … |
| 名詞 m | 0 | 1 | 0 | … | 0 |

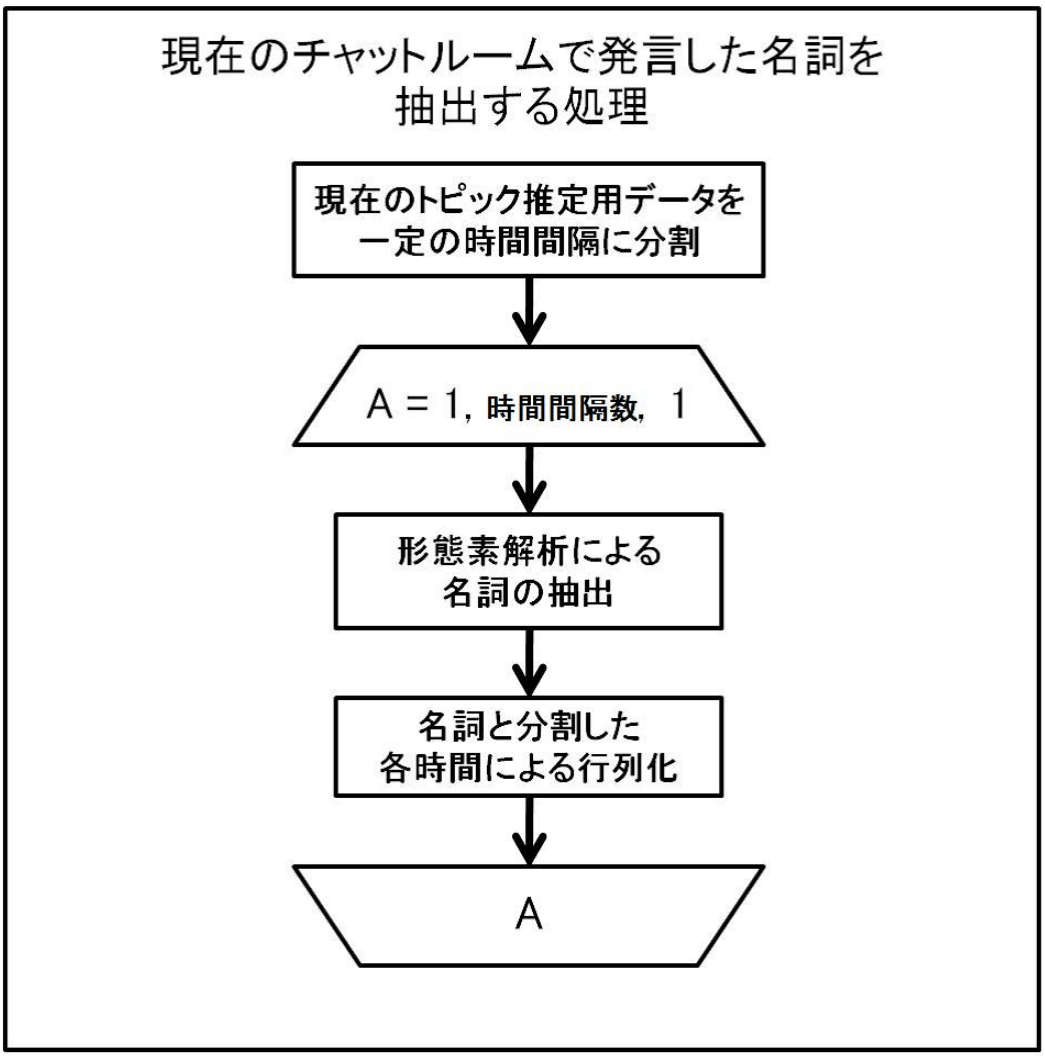


図 4.4: 現在のチャットルームで行われた会話の抽出処理

### 4.3.4 各抽出データのマッチング

この処理は各ユーザの特徴抽出とログの抽出をマージしており、図 4.5 のように処理される。この処理によって、各ユーザの特徴とログの特徴を現在のチャットルームのログに取り入れたデータができる。

この流れ図ではまず、抽出した名詞をトピックとなる名詞群へ分類し、行列データの行項目を縮小する。抽出した名詞をトピックに分類し行列データを縮小することで、トピックの推定にかかる処理時間を大幅に短縮できる。

次に図 4.6 のアルゴリズムを行うことで、現在のチャットルームに参加している各ユーザの過去のログの特徴を付加する。

この処理項目について、図 4.7 を使って説明する。この図は現在のチャットルームにおいて、各ユーザの発言した名詞数が各ユーザの過去の発言履歴データを適用することで、どのように変化するかを示している。

たとえば、トピック C に図 4.6 のアルゴリズムを適用する。
まず、ユーザ A の変化は、以下の式となる。

$$2.0 + 0/100 * 10 * 0.1 * 0.5 = 2.0 \tag{4.1}$$

次に、ユーザ B の変化は、以下の式となる。

$$0 + 20/50 * 10 * 0.9 * 0.5 = 1.8 \tag{4.2}$$

よって、過去の発言履歴データ適用後のトピック C の要素は、式 4.1 と式 4.2 を加えた 3.8 となる。つまり、式 4.1 と式 4.2 によって過去の発言履歴データを適用することで、2 だったトピック C の名詞数を 3.8 として扱うこととなる。このような計算をすべてのトピックに適用することで、現在のチャットルームに参加している各ユーザの過去のログの特徴を付加できる。

最後に図 4.8 のアルゴリズムを行うことで、現在のチャットルームに直前のログの特徴を付加する。

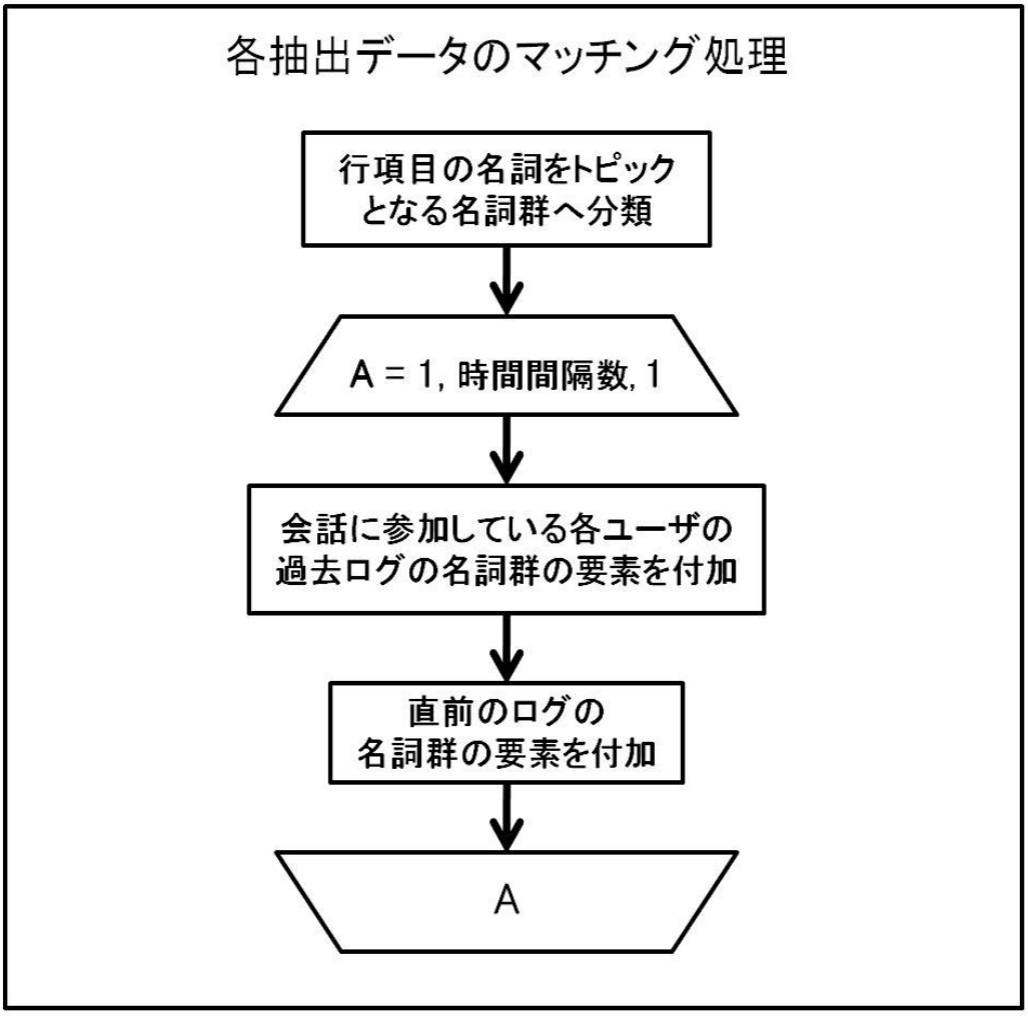


図 4.5: 各抽出データのマッチング処理

```
for( i = 0 ; i < column ; i++) {
    for( j = 0 ; j < user[i] ; j++) {
        for( k = 0 ; k < topic ; k++) {
          now[i , k] += now[i , j , k] + user-noun[i , j, k] / user-all-noun[i , j]
                        * all-now-noun[i] * remark-ratio[i , j] * user-effect ;
        }
    }
}
```

column:一定の時間間隔に分割した分割数
user[i]:時間iの発言ユーザ数
topic:トピックの数
now[i , k]:時間iのログに含まれるトピックkの名詞数
now[i , j , k]:時間iでの、ユーザjのログに含まれるトピックkの名詞数
user-noun[i, j, k]:時間iでの、ユーザjのログに含まれるトピックkの名詞数
user-all-noun[i, j]:時間iでの、ユーザjのログに含まれる名詞数
all-now-noun[i]:時間iのログに含まれる名詞数
remark-ratio[i, j]:時間iでの、ユーザjの発言比率
user-effect:過去の発言履歴データの適用率

図 4.6: 各ユーザの過去のログの特徴を付加するアルゴリズム

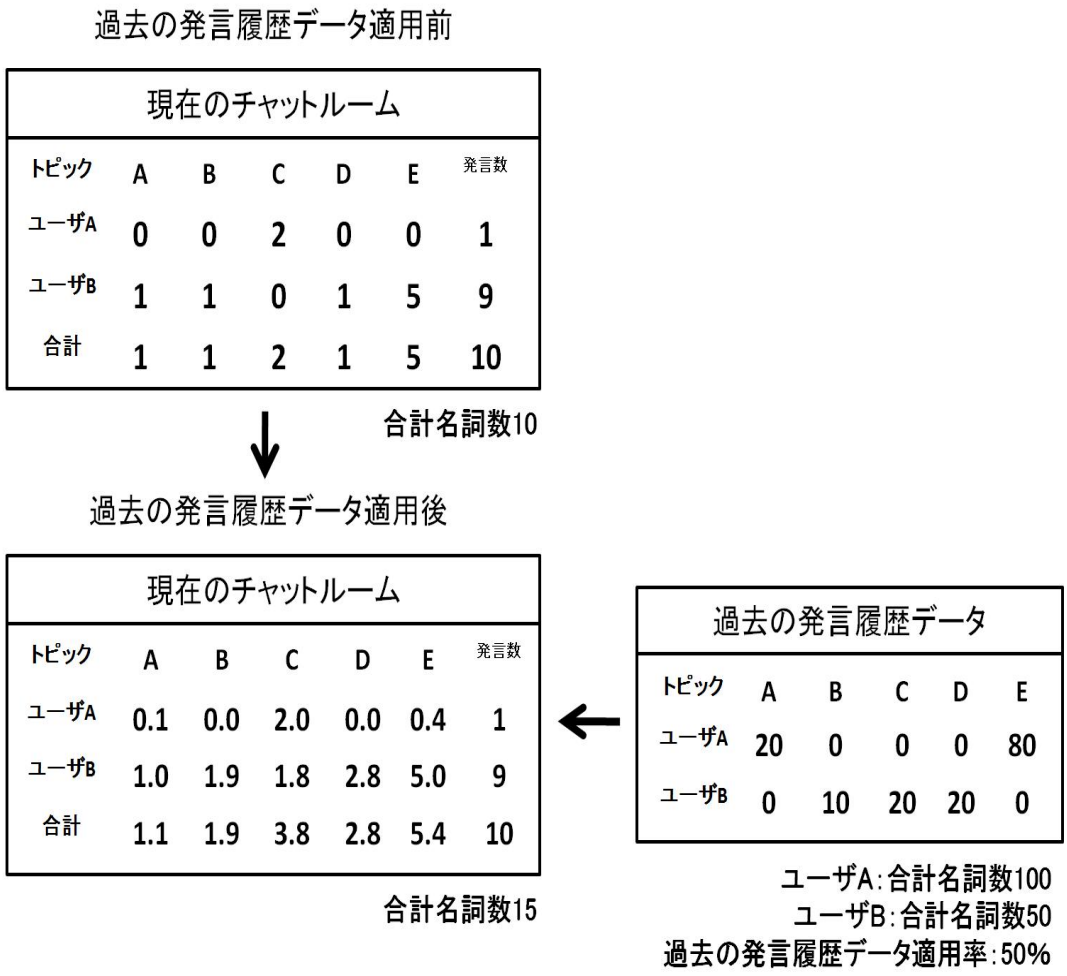


図 4.7: 各ユーザの過去の発言履歴データの適用例

```
for( i = 1 ; i < column ; i++) {
      for( j = 1 ; j < topic ; j++) {
           now[i , j] += now[i-1 , j] * just-before-effect;
            }
      }
}

column:一定の時間間隔に分割した分割数
topic:トピックの数
now[i , j]:時間iのログに含まれるトピックjの名詞数
just-before-effect:直前ログの適用率
```

図 4.8: 直前のログの特徴を付加するアルゴリズム

この処理項目について、図 4.9 を使って説明する。この図は現在のチャットルームにおいて、各ユーザの発言した名詞数が各ユーザの過去の発言履歴データを適用することで、どのように変化するかを示している。たとえば、トピック E に図 4.8 のアルゴリズムを適用すると以下の式のようになる。

$$7 + 8 * 0.5 = 11 \tag{4.3}$$

よって、直前のログ適用後のトピック E の要素は式 4.3 から 11 となる。つまり、式 4.3 によって直前のログを適用することで、7 だったトピック E の名詞数を 11 として扱うこととなる。このような計算をすべてのトピックに適用することで、現在のチャットルームに直前のログの特徴を付加できる。

### 4.3.5 現在のチャットルームのトピックの推定

この処理では各トピックに属する名詞の偏りを測ることで、現在のチャットルームのトピックを推定する。各ユーザの特徴とログの特徴を現在のチャットルームのログに取り入れたデータから、トピック推定を行うため、少ない情報量のログの意味を拡大し、特徴の捉え難いものの特徴も取り入れられる。よって、チャットに対して最適なトピックを抽出できる。この処理は、図 4.10 のように処理される。

この流れ図ではまず、図 4.11 のアルゴリズムを行うことで、各時間のトピックの分散を計算する。このアルゴリズムは名詞の偏りを大きく拡大するため、トピックをうまく推定ことができる。

この処理項目について、図 4.12 を使って説明する。この図は、トピックの分散値の算出例を示している。この例では、トピック B の分散値が最大分散値となる。このような計算を各時間において行うことで、トピックを推定することができる。

次に、最大分散値がトピックの基準となる規定値を超えているかを判定する。抽出された名詞の偏り、即ち最大分散値が小さければ、どのトピックに対しても特徴

的ではないのでトピックとはいえない。これを反映し、最大分散値が基準値を超えていれば、発言数が規定値以上を満たしているかを測る。これによって、会話として成り立っていないと考えられる多くの名詞を含む一つの長い発言だけのログでも、トピックとしてしまうようなことを回避できる。

そして最後に、発言数が規定値以上を満たしていれば、トピックとその時の時間と最大分散値を格納し、格納したトピックを表示する。また、最大分散値が基準値を超えていなければ、この時間はトピックが存在しないとして、次の時間のトピック推定の処理を行う。

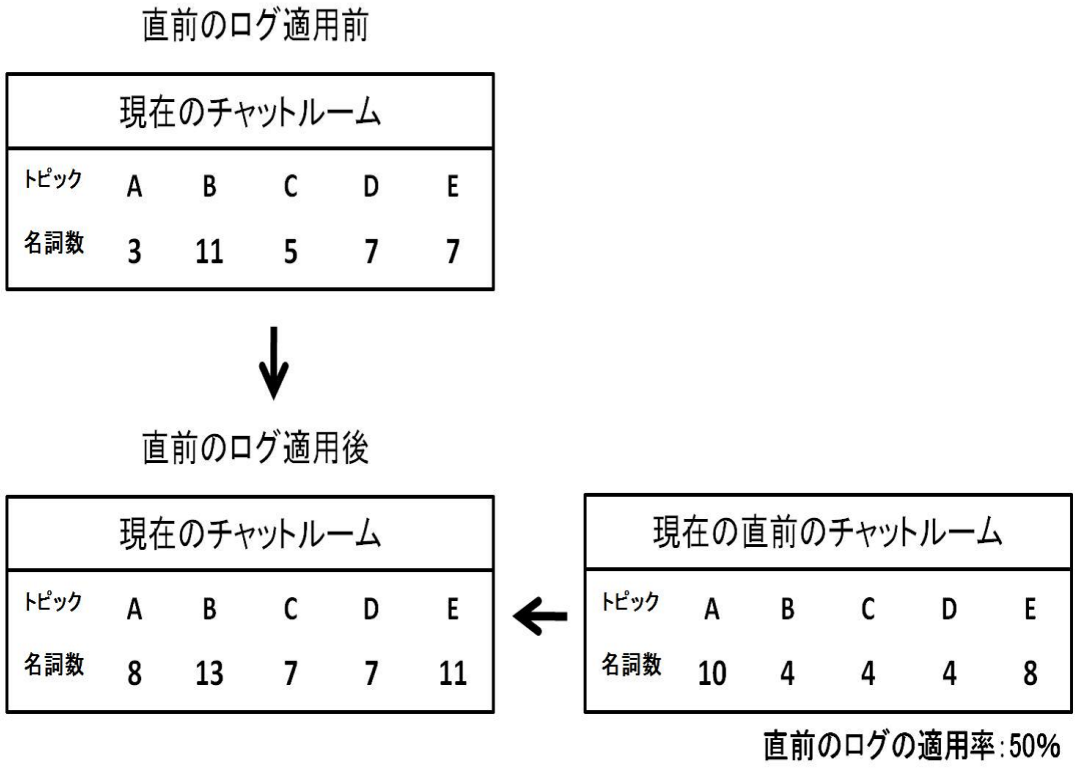


図 4.9: 直前のログの特徴の適用例

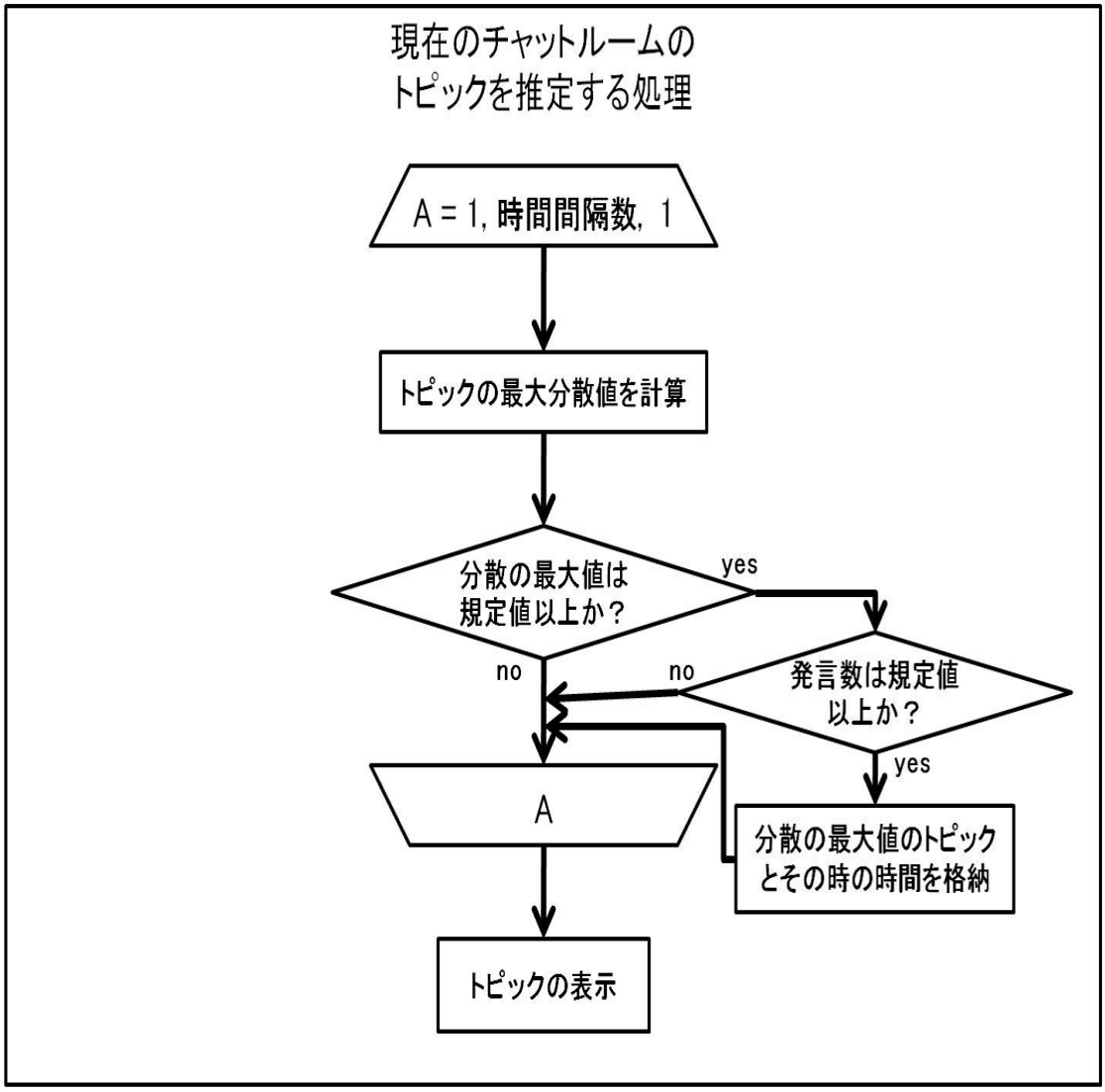


図 4.10: 現在のチャットルームのトピックの推定処理

```
for( i = 0 ; i < column ; i++) {
    for( j = 0 ; j < topic ; j++) {
        ave[i] += now[i , j] / topic;
    }
    for( j = 0 ; j < topic ; j++) {
        if(now[i , j] >= ave[i]) {
            var[i , j] = (now[i , j] - ave[i])^2 ;
        }else{
            var[i , j] = - (now[i , j] - ave[i])^2 ;
        }
    }
    max-var[i] = var[i , 0] ;
    for( j = 1 ; j < topic ; j++) {
        if(var[i , j] >= max-var[i] {
            max-var[i] = var[i , j] ;
        }
    }
}
```

column：一定の時間間隔に分割した分割数
topic：トピックの数
ave[i]：時間iの1トピック当たりの平均名詞数
now[i , j]：時間iのログに含まれるトピックjの名詞数
var[i , j]：時間iのトピックjの分散値
max-var[i]：時間iの最大分散値

図 4.11: 各時間のトピックの分散を計算するアルゴリズム

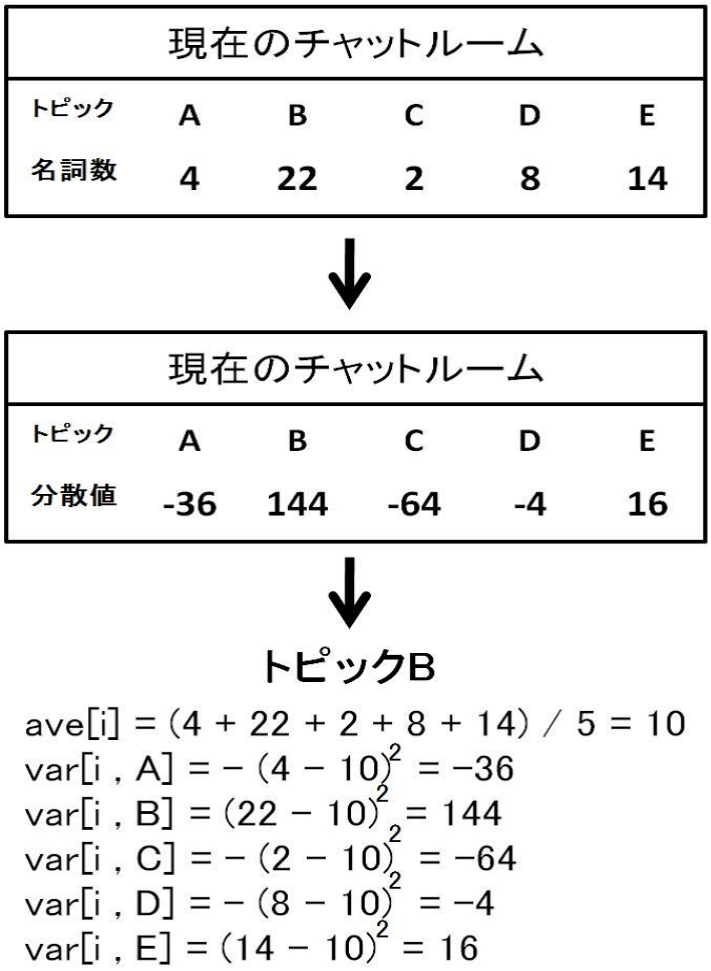


図 4.12: トピックの分散値の算出例

# 第 5 章

# 実装

本章では、本研究で提案するシステムの実装について説明する。

実装には、Perl 5.10.0[6] と R 2.9.0[7] を使用した。

## 5.1 各ユーザが過去に発言した名詞を抽出

この処理は以下のように実装する。

1. 各チャットルームのログの各行に含まれているユーザネームから、ユーザごとにログをテキストファイルに分割する。
2. R のパッケージ RMeCab[8] を用いて、ユーザごとに分割したテキストファイルに形態素解析を行い、名詞を抽出する。

RMeCab　RMeCab とは、石田が開発した R のパッケージである。これよって、R から形態素解析器 MeCab[9] を解析させることができる。またその結果を、R で標準的なデータ形式に変換して出力することができる。

### 5.1.1 名詞のデータクリーニング

形態素解析器によって抽出された名詞は完全ではないため、データクリーニングをする必要がある。本研究ではデータクリーニングの処理として、抽出された名詞に以下の処理を行った。

- 1 文字の漢字の名詞を削除する。
- 2 文字以下の平仮名、片仮名の名詞を削除する。
- 1 文字目が小さい平仮名、小さい片仮名の名詞を削除する。
- 記号、数字を含む名詞を削除する。
- 半角文字を全角文字に修正する。

このような処理を行うと、本来トピックになりえる名詞も削除してしまう可能性がある。たとえば、この処理では「パン」「米」「魚」などの名詞を削除してしまうことになる。このような名詞は、「食べ物」のトピックとしては重要な名詞であると言える。しかし、このような処理を行わないとノイズとなる名詞が多くなってしまい、トピックをうまく推定できなくなるため、データクリーニングをする必要がある。

## 5.2 現在の会話の直前に行われた会話の名詞の抽出

この処理は以下のように実装する。

1. 各チャットルームのログの各行に含まれているタイムスタンプから、5 分間隔ごとにこのログをテキストファイルに分割する。
2. R のパッケージ RMeCab を用いて、5 分間隔ごとに分割したテキストファイルに形態素解析を行い、名詞を抽出する。

## 5.3 現在のチャットルームで行われた会話の名詞の抽出

この処理は以下のように実装する。

1. ログの各行に含まれているタイムスタンプから、5 分間隔ごとにログをテキストファイルに分割する。

2. R のパッケージ RMeCab を用いて、5 分間隔ごとに分割したテキストファイルに形態素解析を行い、名詞を抽出する。

3. R のパッケージ RMeCab を用いて、行項目に形態素解析を行った名詞、列項目に 5 分間隔ごとの時間をとり、行列化したデータを作る。

## 5.4 各抽出データのマッチング

この処理は以下のように実装する。

1. 「分類語彙表増補改訂版」データベース [10]、Yahoo!デベロッパーネットワークの Yahoo!検索 Web API[11] を用いて、行列の行項目の名詞をトピックとなる名詞群に分類する。

2. 各時間のログに対して、会話に参加している各ユーザの過去のログの特徴を図 4.6 のアルゴリズムを行うことで付加する。

3. 現在のチャットルームに図 4.8 のアルゴリズムを行うことで、直前のログの特徴を付加する。

このアルゴリズムのパラメタは、$column$ を 2016、$topic$ を 5、$just\text{-}before\text{-}effect$ を 0.5 とした。パラメタの値については、$column$ は実装したシステムで処理したトピック推定用データが一週間分のログであったが、この一週間のログを 5 分間隔に分割すると、分割数が 2016 となるためである。$topic$ については、トピックの推定精度の実用性を考慮して、5 に固定したためである。$just\text{-}before\text{-}effect$ については、経験則的に 0.5 程度反映させることが妥当だとした。この処理の結果、5 分間隔の各チャットログに、5 種類のトピックの名詞がそれぞれ、どれだけ含まれているかを示すデータができる。

### 5.4.1 名詞のトピックへの分類

本実装では、Perl、「分類語彙表増補改訂版」データベース、Yahoo!デベロッパーネットワークの Yahoo!検索 Web API を用いて名詞のトピックへの分類を行う。

**「分類語彙表増補改訂版」データベース** 分類語彙表とは、語を意味によって分類、整理したシソーラス（類義語集）である [10]。またこれは、研究開発用のデータベース版である。レコード総数は 101,070 件あり、数多くの語を含んでる。

Yahoo!**デベロッパーネットワーク** Yahoo!デベロッパーネットワークとは、ソフトウェア開発者が Yahoo! JAPAN のコンテンツやサービス、技術を利用して、新しいアプリケーションを作成するための Web サービスを無料で提供しているサービスである [11]。

本実装では、各名詞とトピックの関連性を測るためにこのサービスのウェブ検索 API を利用する。

この処理は以下のように実装する。

**処理 A** 「分類語彙表増補改訂版」データベースの名詞の分類項目を参考に、トピックを「植物、食料」「身体、生命」「経済、公私」「芸術」「機械」の 5 つにわける。

**処理 B** Perl を用いて、「分類語彙表増補改訂版」データベースの中の 5 つのトピックに属する名詞に各抽出データの名詞が一致するならば、各抽出データの名詞を 5 つに分類する。

**処理 C** 処理 B によって分類できなかった各抽出データの名詞を対象に、Perl と Yahoo!デベロッパーネットワークの Yahoo!検索 Web API を用いて、各名詞とトピックとの関連性を測る。関連性を測るために、抽出された各名詞と図 5.1 の各トピックの代表名詞の組合せをクエリとした検索結果を得る。検索

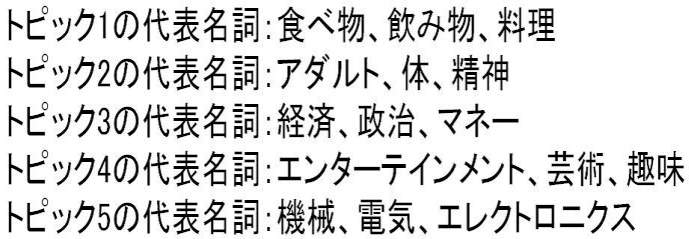


図 5.1: 5 つのトピックの代表名詞

結果の数が 50000 件未満のものは、意味の理解できない名詞も多いため除外する。

**処理 D** 処理 C の検索結果の数を用いて、まだ分類できていない名詞と図 5.1 の 5 つのトピックの各代表名詞とのシンプソン係数を計算する。各トピックにおいて、各代表名詞との最大のシンプソン係数をそのトピックとの関連性の指標とする。

**シンプソン係数** シンプソン係数とは、自然言語処理における係数の一種で、X というキーワードと Y というキーワードが同じページや同じ文書内で出現する（共起する）場合の頻度の強さを表現する。シンプソン係数は、以下の式で表される [12]。

$$simp(X,Y) = \frac{|X \cap Y|}{min(|X|,|Y|)} \tag{5.1}$$

本実装では、これを同じチャットルームのログに出てくる名詞とトピックの関連性を測るために用いた。

**処理 E** 各名詞と 5 つのトピックとのシンプソン係数に図 5.2 のアルゴリズムを行い、各抽出データの名詞をトピックに分類する。

```
for( i = 0 ; i < noun-num ; i++) {
    for( j = 0 ; j < topic ; j++) {
        simp-ave[i] += simp[i , j] / topic ;
    }
    for( j = 1 ; j < topic ; j++) {
        if((max-simp[i] > 0.300) && (max-simp[i] > (simp-ave[i] * 2))) {
            topic-noun[j] = noun[i] ;
        }
    }
}
```

noun-num：処理Bによって分類できなかった名詞数
topic：トピックの数
simp-ave[i]：名詞iの1トピック当たりの平均のシンプソン係数
simp[i , j]：名詞iでの、トピックjのシンプソン係数
max-simp[i]：名詞iでの、最大のシンプソン係数
noun[i]：処理Bによって分類できなかったi番目の名詞
topic-noun[i]：トピックとして分類する名詞i

図 5.2: 名詞のトピック分類アルゴリズム

## 5.5 現在のチャットルームのトピックの推定

この処理は以下のように実装する。

1. 5 種類のトピックの名詞がどれだけ含まれているかを示すデータの各時間に対して、図 4.11 のアルゴリズムを行う。

2. 図 4.11 のアルゴリズムを行った結果の各時間の最大分散値が、トピックとなる基準値を超えているかを判定する。経験則的に最大分散値が 50 を超えるログには、トピックが存在してることが多い。よって、基準値 50 を超えていればこの時間のトピックと見なし、超えていなければこの時間はトピックが存在しないこととする。

3. 経験則的に発言数が 10 を超えるログには、複数ユーザの会話によるトピックが存在してることが多い。よって、トピックと見なされたデータに対して、各時間での発言数は 10 以上であるかを判定し、10 以上であれば、この時間のトピックとして見なし、トピックの出力のために、トピックが存在した時間、トピックの種類、最大分散値を格納する。発言数が 10 を超えていなければ、この時間はトピックが存在しないこととする。

4. トピックとして判定されたデータを出力する。

## 5.6 正しいトピック推定と間違ったトピック推定

実装のシステムによって、現在のチャットルームのトピックの推定を行った。以下で、正しいトピック推定と間違ったトピック推定を実際のチャットログの例を用いて説明する。

### 5.6.1 正しいトピック推定

図 5.3 は、正しいトピック推定の例である。このログは、トピック推定では政治、経済のトピックと推定された。また目視でのトピック付けでも、株、銀行、政党等の名詞を交えた会話を行っているので、政治、経済のトピックとした。

### 5.6.2 間違ったトピック推定

図 5.4 は、間違ったトピック推定のログの例である。このログは、トピック推定では食べ物のトピックと推定された。しかし、目視トピック付けでは京都の地名や観光名所の話が中心で旅行に近い話だと判断し、趣味、芸術、エンターテインメントのトピックとした。

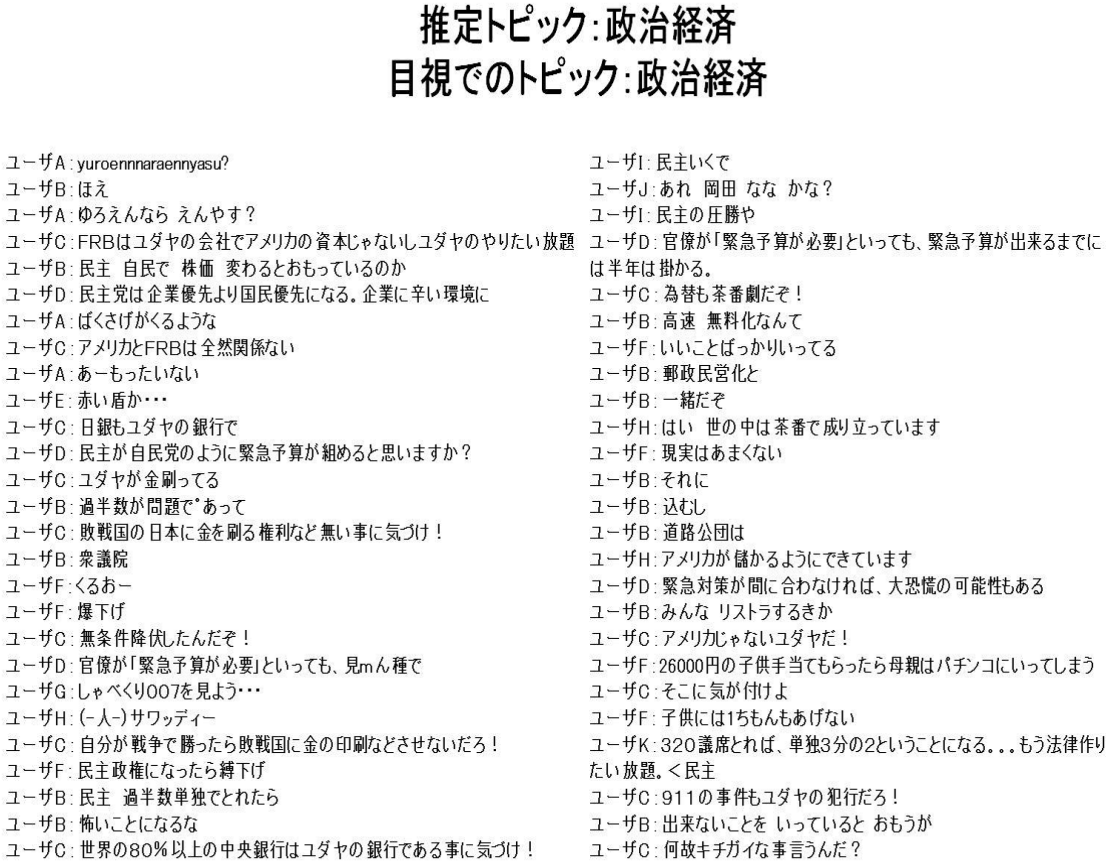


図 5.3: 正しくトピック推定されたログの例

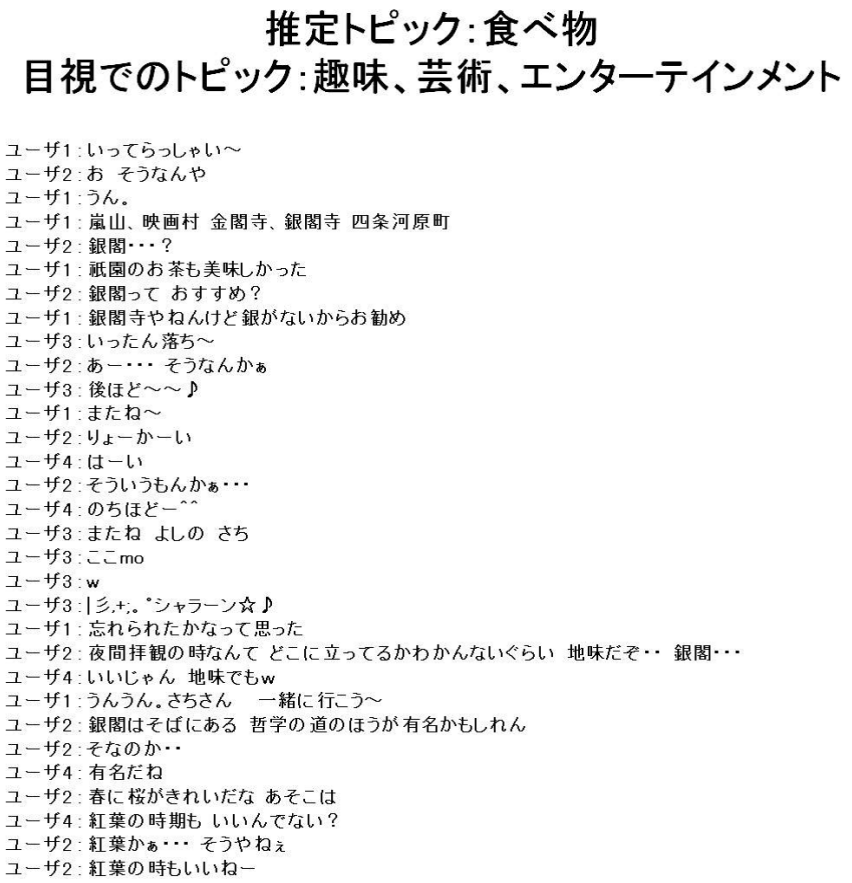


図 5.4: 間違ってトピック推定されたログの例

# 第 6 章

# 実験

本章では実際のチャットログを用いて現在のチャットルームのトピックの推定を行い、本システムの有効性を測る。

実験では、トピックを予め決めた5つのトピックだけとした。推定トピックはそのトピックの中から選ばれることとした。トピックは「食べ物」「アダルト、身体」「政治、経済」「趣味、芸術、エンターテインメント」「テクノロジ」の5つとした。

## 6.1 実験方法

実験は、以下の2つを行う。実験1は、本研究の手法のトピックの推定精度を測る目的で行う。実験2は、本研究の手法のトピックの継続性を測る目的で行う。

**実験**1　この実験は、各ユーザの特徴、直前のログの特徴、トピック抽出の計算方法の効果を調べることを目的として行う。実験方法は、実際の7つのチャットルームからそれぞれ、トピックとして推定したログの最大分散値の値が最も高い上位30位までの推定トピックを取り出す。そして、推定トピックのログを目視によってつけたトピックを正解として、推定トピックの有効性を調べる。

**実験**2　実装したシステムの有効性の指標の一つは、トピックを目安にチャットルームに参加し、そのトピックについての会話を行うことができるかどうかということだと考えられる。よって、トピックについての会話を行うには継続性のあるトピッ

クでなくてはならない。この実験は、直前のログの特徴によってトピックの継続性の効果を調べることを目的として行う。

実験方法は、実験 1 で取り出された推定トピック、推定トピックのログの直後のログを取り出す。直後のログとは、次の 5 分間のログを指す。そしてこの直後のログに、目視によってつけたトピックを正解として、推定トピックの継続性を測る。

## 6.2 評価方法

実験 1 の評価方法として、現在のチャットルームのログに対して、トピックの上位 30 位までの適合率と上位 10 位までの $MAP$(Mean Average Precision) 値を測った。

適合率とは、推定トピックが目視でログにつけたトピックと一致している割合である。

この値が高いほど推定トピックが正しい割合が多く、有効性が高いと言える。$MAP$ とは、以下の式で表される値である。

$$MAP = \frac{\sum_{m=1}^{M}\{\sum_{r=1}^{R}(\frac{h(r)}{r} * H(r))\}}{M} \tag{6.1}$$

$M$ はトピックを抽出するチャットルームの数、$R$ は各チャットルームの式 6.1 の上位 R 位の数、$h(r)$ は各チャットルームの式 6.1 の r 位までの正しい推定トピックの数、$H(r)$ は各チャットルームの式 6.1 の r 位の推定トピックが正しければ 1、間違っていれば 0 になる数である。この値が高いほど、信頼性の高い推定トピックが、より正しいということになり、有効性が高いと言える。

実験 2 の評価方法として実験 1 で推定したログの直後のログに対して、トピックの上位 30 位までの適合率を測った。この値が高いほど、現在のチャットルームの直後のログの継続性があるため、有効性が高いと言える。

比較対象として、以下の手法を挙げる。実験 1 は、各ユーザの過去の特徴の有無 (user effect)、直前のログの特徴の有無 (just before)、2 種類のトピック抽出法

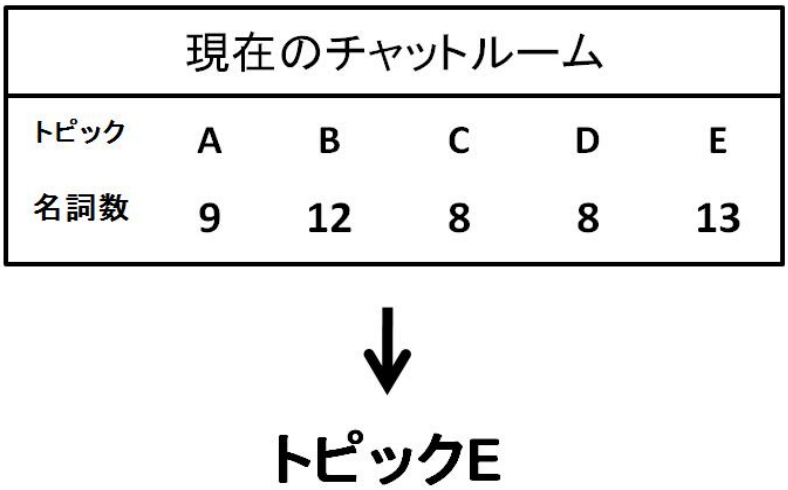


図 6.1: 単純最大手法の例

(algorithm)(本手法 (v) と単純最大手法 (m)) の組合せを比較する。

**単純最大手法**　単純最大手法とは、現在のチャットルームのログにおいて抽出された名詞の中で、最大の名詞数が属するトピックのカテゴリをトピックとする方法である。

たとえば図の例では、現在のチャットルームのログにおいて、トピック E が 5 つのトピック中では最大の 13 個の名詞が属しているので、このチャットルームではトピック E をトピックとする。

具体的には、以下の 5 つの手法を比較する。

**手法 1(本研究の手法)** この手法は本研究が採用する手法で、各ユーザの過去の特徴と直前のログの特徴を取り入れ、最大分散値を計算することによって、トピックを抽出する。

**手法 2** この手法は直前のログの特徴だけを取り入れ、最大分散値を計算することによってトピックを抽出する。

**手法 3** この手法は各ユーザの過去の特徴だけを取り入れ、最大分散値を計算することによってトピックを抽出する。

**手法 4** この手法は各ユーザの過去の特徴と直前のログの特徴は取り入れず、最大分散値を計算することによってトピックを抽出する。

**手法 5** この手法は各ユーザの過去の特徴と直前のログの特徴は取り入れず、単純最大手法によってトピックを抽出する。

実験 2 では、直前のログの特徴の有無 (just before) について比較する。

具体的には、以下の 2 つの手法を比較する。

**手法 1(本研究の手法)** この手法は本研究が採用する手法で、各ユーザの過去の特徴と直前のログの特徴を取り入れ、最大分散値を計算することによってトピックを抽出する。

**手法 2** この手法は直前のログの特徴だけを取り入れ、最大分散値を計算することによって、トピックを抽出する。

## 6.3 チャットログ

実験では、実際の 7 つのチャットルームのログ 11 日分を収集した。7 つのチャットルームのテーマは、政治、大阪、株、中学生、アダルト、30 代、20 代である。この内、現在のトピック推定用データは 7 日分、過去の発言履歴データは 4 日分とした。その結果、現在のトピック推定用データは 420,232 行、過去の発言履歴データは 218,495 行となった。

### 6.3.1 チャットログのデータクリーニング

収集したチャットログはスパムが多いため、取り除く必要がある。以下のものをスパムと見なし、Perl によって削除した。

**スパム 1** URL を含む発言

**スパム 2** 発言者のプロフィールの URL への誘導を行う発言

**スパム 3** 規則性がある発言

このようなスパムは、機械によって自動で行われることが多い。

**スパム** 1　URL を含む発言

URL を含む発言は、発言ユーザの広告サイトへの誘導であることが多い。そのような意図がない発言でも、URL だけを発言することが多い。URL だけの発言から、名詞を抽出することは妥当ではないため、削除しても問題ない。

**スパム** 2　発言者のプロフィールへの誘導を行う発言

このチャットでは、各ユーザは、自分のプロフィールサイトを持っている。発言者がこのプロフィールを見せようとする意図の発言も、発言ユーザの広告サイトへの誘導であることが多い。このような発言者のプロフィールには URL が記載されており、発言ユーザの広告サイトである可能性が高い。

**スパム** 3　規則性がある発言

チャットでは広告サイトへ誘導する意図がなくとも、一定間隔で同じ発言やある単語の組合せのループ等、規則性のある発言をするユーザもいる。このような発言はトピック抽出においてノイズとなりうるため、削除する必要がある。

スパムの除去を行った結果、トピック推定データは 420,232 行から 130,740 行に、過去の発言履歴データは 218,495 行から 69,843 行となった。最終的に残った発言のみのログは、トピック推定データは 7.65MB、過去の発言履歴データは 3.54MB となった。

# 第 7 章

# 実験結果と評価

本章では第 6 章の実験の実験結果を示し、実験結果に対しての評価を議論する。

## 7.1 提案システムの推定結果

本研究で提案した手法を実際のチャットルームに適用した例として、テーマが株式のチャットルームの 1 日目の推定結果を図 7.1 に示す。図は上から、発言数、目視でのトピック、推定トピックの推移を表している。上の折れ線グラフは発言数の推移、5 分当たりの発言数を表す。真ん中の横線は、トピック数の条件を設けずに目視で自由に付けられたトピックを表す。横線の長さは、トピックの会話が行われた時間を指す。横線の横軸の位置は、その会話が行われた時間を指す。下の点グラフは、推定トピックの推移を表す。プロットの大きさは最大分散値の大きさを表し、トピック推定の信頼度を指す。点の横軸の位置は、その会話が行われた時間を指す。点の縦軸の位置は、トピックの種類を指す。

図 7.1 は、主に、政治、経済の会話が行われていると推定されている。目視で付けられたトピックも政治、経済の会話が多く、時間帯も推定トピックとおおよそ一致しており、本研究の実装したシステムが正常に処理されていると考えられる。

「政治」「大阪」「株」「中学生」「アダルト」「30 代」「20 代」の 7 つのチャットルームすべての各時間の発言数、目視でのトピック、推定トピックを示した図は、付録 B で示す。

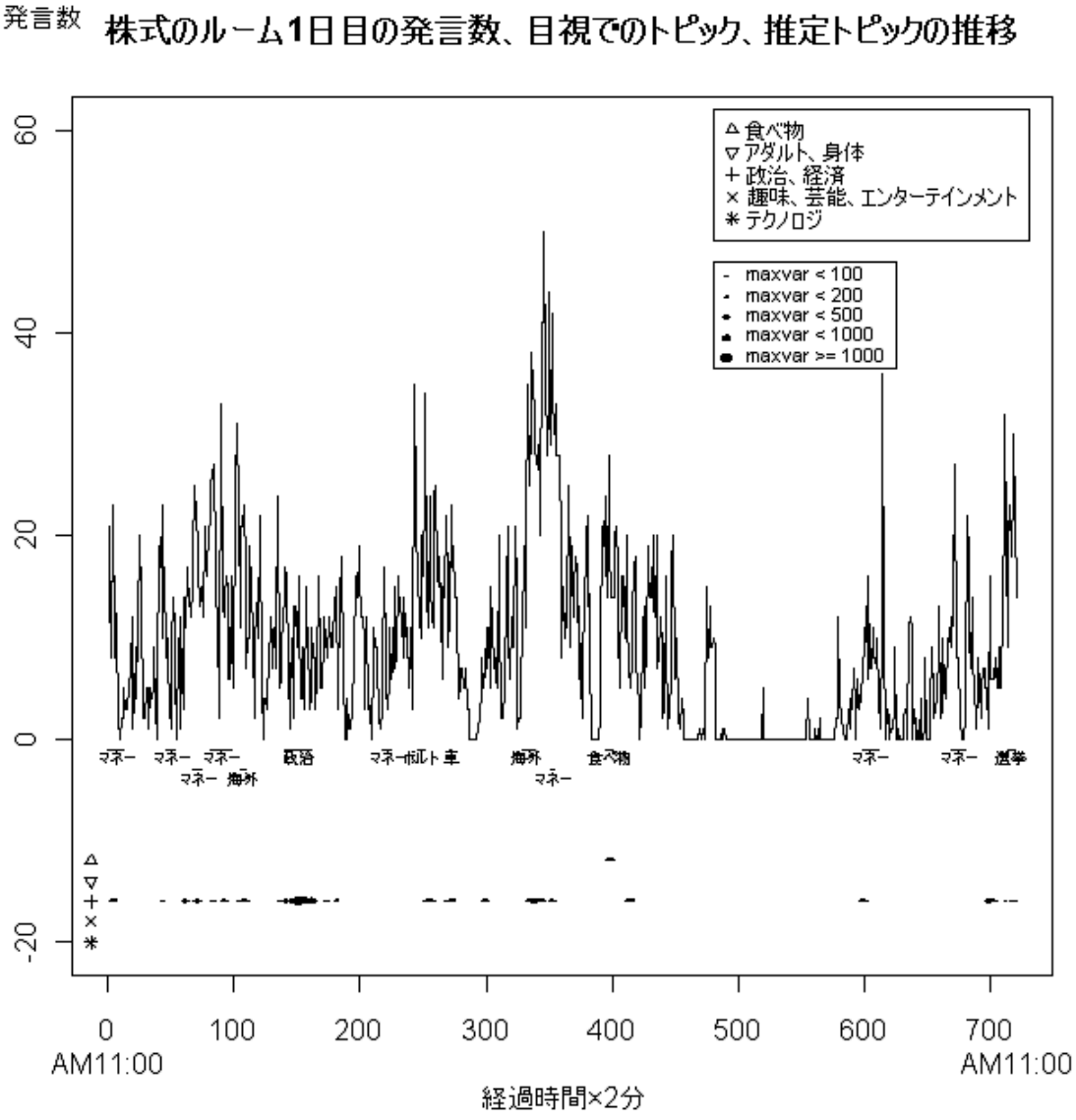


図 7.1: 株式の 1 日目の発言数、目視でのトピック、推定トピックの推移

## 7.2 実験結果

第 6 章の実験 1 の実験結果を、表 7.1 と表 7.2 に示す。表 7.1 は上から、第 6.2 節の比較手法、現在の各チャットルームに対する適合率、現在の全チャットルームの適合率の平均を表している。この値が高いほど推定トピックが正しい割合が多く、本システムの有効性が高いと言える。

表 7.2 は上から、第 6.2 節の比較手法、現在の各チャットルームに対する $MAP$ 値、現在の全チャットルームの $MAP$ 値の平均を表している。この値が高いほど信頼性の高い推定トピックが、より正しいということになり、本システムの有効性が高いと言える。

次に、第 6 章の実験 2 の実験結果を表 7.3 に示す。表 7.3 は上から、第 6.2 節の比較手法、現在のチャットルームの直後のログに対する各チャットルームの適合率、現在のチャットルームの直後のログに対する全チャットルームの適合率の平均を表している。この値が高いほど、現在のチャットルームの直後のログの継続性があるため、有効性が高いと言える。

## 7.3 評価

実験結果全体を通して、適合率の平均、$MAP$ においては、手法による違いよりもチャットルームのテーマの違いによる差の方が大きかった。たとえば、政治、株、アダルトのテーマのチャットルームはそのテーマに関する会話が多いため、トピックも容易に抽出できると考える。一方、大阪、中学生、30 代、20 代のテーマのチャットルームはそのテーマから考えられるトピックが幅広いため、トピックの抽出が難しいと考えられる。

表 7.1: 現在のチャットルームの推定トピックの適合率

| 手法 | 手法 1 | 手法 2 | 手法 3 | 手法 4 | 手法 5 |
|---|---|---|---|---|---|
| user effect | ○ | × | ○ | × | × |
| just before | ○ | ○ | × | × | × |
| algorithm | v | v | v | v | m |
| 政治 | 29/30 | 30/30 | 29/30 | 29/30 | 29/30 |
| 大阪 | 23/30 | 24/30 | 22/30 | 22/30 | 23/30 |
| 株 | 29/30 | 29/30 | 29/30 | 29/30 | 29/30 |
| 中学生 | 22/30 | 22/30 | 23/30 | 23/30 | 23/30 |
| アダルト | 30/30 | 30/30 | 30/30 | 29/30 | 29/30 |
| 30 代 | 25/30 | 24/30 | 26/30 | 27/30 | 27/30 |
| 20 代 | 29/30 | 29/30 | 29/30 | 30/30 | 29/30 |
| 適合率の平均 | 187/210 | 187/210 | 188/210 | 189/210 | 189/210 |
| 適合率の平均 | 0.890 | 0.895 | 0.900 | 0.895 | 0.900 |

表 7.2: 現在のチャットルームに対する推定トピックの MAP 値

| 手法 | 手法 1 | 手法 2 | 手法 3 | 手法 4 | 手法 5 |
|---|---|---|---|---|---|
| user effect | ○ | × | ○ | × | × |
| just before | ○ | ○ | × | × | × |
| algorithm | v | v | v | v | m |
| 政治 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 大阪 | 0.627 | 0.596 | 0.659 | 0.704 | 0.471 |
| 株 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 中学生 | 0.790 | 0.790 | 0.657 | 0.835 | 0.657 |
| アダルト | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 30 代 | 0.790 | 0.835 | 0.790 | 0.790 | 0.790 |
| 20 代 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| MAP | 0.887 | 0.889 | 0.904 | 0.872 | 0.831 |

表 7.3: 現在のチャットルームの直後のログに対する推定トピックの適合率

| 手法 | 手法 1 | 手法 2 |
|---|---|---|
| user effect | ○ | × |
| just before | ○ | ○ |
| algorithm | v | v |
| 政治 | 28/30 | 28/30 |
| 大阪 | 16/30 | 12/30 |
| 株 | 29/30 | 28/30 |
| 中学生 | 12/30 | 11/30 |
| アダルト | 15/30 | 16/30 |
| 30 代 | 20/30 | 20/30 |
| 20 代 | 20/30 | 19/30 |
| 平均適合率 | 140/210 | 134/210 |
| 平均適合率 | 0.667 | 0.638 |

### 7.3.1 実験 1 の評価

実験 1 の適合率の平均においては、どの手法にも大きな差は見られなかった。

$MAP$ 値においては、手法 1、手法 2、手法 3、手法 4 には大きな差が見られなかった。しかし手法 5 は、平均適合率においては、若干ながら他の手法より高かったにも関わらず、$MAP$ 値においては他の手法より劣っている。これは単純最大手法が、信頼度の高い推定トピックを間違えやすいということがわかる。

各ユーザの過去の特徴 (user effect)、直前のログの特徴 (just before) については、大きな効果は表れなかった。しかし本研究の手法では、各ユーザの過去の発言履歴を各ユーザの過去の特徴として考慮しているため、他の手法より抽出したトピックを話す土壌が整っていると考えられる。つまり、抽出されたトピックを目安にチャットルームに行ったとしても、より支障なく会話を行いやすいということができると考えられる。

### 7.3.2 実験 2 の評価

実験 2 の平均適合率においては、手法 1 が高い結果となった。よって、直前のログの特徴 (just before) を取り入れた手法の方が、より継続性のあるトピックが抽出できた。よって、本手法のトピックは継続性があるため、抽出されたトピックを目安にチャットルームに参加すれば、そのトピックの会話を行える可能性が、より高いと考えられる。

# 第 8 章

# 議論

この章は本システムの問題について説明し、その解決法について議論する。

推定されたトピックのログや実験結果から、本研究の問題は主に以下の 2 つが考えられる。

- 抽出された名詞のトピックへの分類の問題
- トピックの推定方法の問題

## 8.1 名詞のトピックへの分類の問題

本研究では、名詞を 5 つのトピックに分類し、チャットルームのトピックを、その 5 つのトピックの中から抽出している。この名詞のトピックへの分類に関して、以下の問題がある。

**問題** 1　トピックの数が少なく、予め固定されている。
様々なテーマが存在するチャットルームでは、多種多様なトピックの会話が行われている。しかし本研究では、固定した 5 つのトピックの中から、トピックを抽出するという方法を取ってしまっている。よってこの方法では、多種多様なトピックが存在するチャットルームをうまく反映できていないと考えられる。

この問題が起こる原因は、名詞の分類方法と抽出されたトピックの実用性にある。

**原因 1-1** 名詞の分類方法

名詞の分類方法については、抽出された各名詞と図 5.1 の各トピックの代表名詞とのシンプソン係数によって関連性を測っていた。しかし、シンプソン係数には、比較する名詞のクエリに対する検索結果の数に、大きな差がある場合は、関連性がうまく算出されない可能性があり、関連性に関して、完全な解を出せない。また、各代表名詞にも、検索結果の数や概念的な階層の違いがあり、抽出された名詞のトピックへの分類が、偏ってしまっているということもある。

**原因 1-2** 抽出されたトピックの実用性

本研究の現在の手法は、固定した 5 つのトピックの中からトピックを抽出するという方法を取っている。このトピック数で実験を行った時は、5 種類という抽象性があるが、高精度にトピックを抽出することができ、推定結果を一つの目安にすることができる。しかしこのトピック数を増やしたり、上限を設けないような方法をとれば、本研究の現在の名詞の分類方法が不完全なこと [原因 1-1] や現在のトピック抽出方法では十分な精度は得られないと考えられる。

**問題** 2　名詞の分類に、時間がかかる。

本研究の名詞のトピックへの分類方法の一つに、Yahoo!デベロッパーネットワークの Yahoo!検索 Web API に対して、抽出された各名詞と各トピックの代表名詞の組合せクエリとした検索結果を得るという処理がある。この処理を実際の 7 つのチャットルーム 7 日分のログの名詞に対して行うと、数百時間かかってしまう。このような膨大な時間がかかってしまうと、ユーザはトピックを目安にチャットルームの会話に参加することができなくなってしまう。このような問題が起きる原因は、外部との通信を行う処理を含んでいるためである。

### 8.1.1 名詞の分類方法の問題に対する議論

本研究の最大分散値を求めるトピック抽出方法は、各トピックにある程度均等に名詞が分類されているということが望まれる。しかし実際には、あるトピックへ名詞が偏ってしまっている。これに対して、属する名詞が少ないトピックに重み付けを行うということも考えられる。しかし、シンプソン係数を用いた名詞分類方法は完全ではないため、重み付けを行うことで正しくない分類が行われた名詞のノイズによる影響が無視できなくなってしまい、正常なトピック抽出が行えないと考えられる。

このことから、問題1の解決に取り組むためには、各トピックに名詞がある程度均等に分類されるような代表名詞を選ばなければならない。これには、言語に関する知識が必要となるだろう。また、最大分散値を求めるトピック抽出方法を、見直すということも必要かもしれない。

### 8.1.2 名詞の分類時間の問題に対する議論

本研究のシステムの実用性を考えると、チャットルームの各時間のログを5分間隔に切っていることから、各時間のトピック抽出処理は抽出したトピックを目安に実際の会話を行うとすると、数十秒から数分以内に行われることが望ましい。また、最大分散値が大きいトピックのログは信頼性の高いトピックだと推定されるが、そのログに含まれる名詞の数は最大で数百個にもなる。数百個の名詞をトピックに分類するとすれば、数十分の時間がかかってしまう。これでは、トピックを抽出した頃にはそのトピックが既に終わってしまっている可能性が高い。

そこでシステムが実用的な時間で処理できるためには、予め、大量の名詞についてトピックの分類分けを行っておく必要がある。そして、新しく出現した名詞だけに限ってトピックの分類を行えば、抽出したトピックを利用することができる。これによって外部との通信を行う処理が縮小され、処理時間が減る。

## 8.2 トピック抽出方法の問題

本研究のトピック抽出方法は、第4章の様々な処理から構成されている。このトピック抽出方法の問題として、以下の5つが考えられる。

**問題A** トピック抽出を一定の時間間隔のログで扱うことで、本来トピックとなるはずだったトピックが抽出されないことがある。

**問題B** 最大分散値を求める現在のトピック推定法は、複数のトピックの混在するトピック抽出ができない。

**問題C** 直前のログの特徴によって、直前のトピックに現在のトピックが引きずられて、別のトピックが抽出されにくいことがある。

**問題D** 各ユーザの過去のログの特徴によって、各ユーザの過去の発言に現在のトピックが引きずられて、別のトピックが抽出されにくいことがある。

**問題E** 1つの長い発言にトピックが引きずられて、トピックが正常に抽出できないことがある。

**問題A** トピック抽出を一定の時間間隔のログだけに行うことで、本来トピックとなるはずだったトピックが抽出されないことがある。
本研究のシステムでは、チャットルームのログを5分間隔で分割し、分割した5分ごとのログに対してトピック抽出を行っている。この方法では、5分間隔の境界を丁度含むような短いトピックが抽出できない。逆に、長時間の連続して会話が行われているトピックを複数抽出してしまうこともある。

この問題の起きる原因は、会話のトピックとは静的な時間間隔で行われるわけではないが、本研究では短い静的な時間間隔のログだけをトピック抽出の対象としているためだと考えられる。

**問題**B　最大分散値を求める現在のトピック推定法は、複数のトピックの混在するトピック抽出ができない。
本研究のトピックの推定は、図 4.11 のアルゴリズムを行うことによって算出される。しかし、このアルゴリズムでは、複数のトピックが混在していた時、最大分散値をより小さくしてしまう。更に、トピックとなりづらいばかりか、トピックとして算出されたとしても、どちらか一方だけしか推定できない。

この問題の起きる原因は、図 4.11 のアルゴリズムは、このアルゴリズムの性質上、単一のトピックの抽出だけにしか適していないためである。

**問題**C　直前のログの特徴によって、直前のトピックに現在のトピックが引きずられて、別のトピックが抽出されにくいことがある。
本研究では直前のログの特徴を取り入れることによって、より継続性のあるトピックを抽出することができた。しかし、現在のトピックが別のトピックに変わろうとしていても、直前のトピックに現在のトピックが引きずられてしまい、うまくトピック抽出を行えない可能性がある。

この問題の起きる原因は、トピックの変わり目をうまく認識できていないことであると考えられる。

**問題**D　各ユーザの過去のログの特徴によって、各ユーザの過去の発言に現在のトピックが引きずられて、別のトピックが抽出されにくいことがある。
本研究では 各ユーザの過去のログの特徴を取り入れることによって、より支障なく会話が行えるであろうトピックを抽出することができた。しかし、現在のチャットルームのトピックが、そのチャットルームでは馴染みのないトピックの会話を行っていたとしても、各ユーザの過去の発言に現在のトピックが引きずられてしまい、馴染みのないトピックは抽出されない可能性がある。

この問題の起きる原因は問題 C と同様に、トピックの変わり目をうまく認識できていないことであると考えられる。

**問題**E　1つの長い発言にトピックが引きずられて、トピックが正常に抽出できないことがある。
本研究のトピックとなる名詞は、チャットルームに含まれるすべてのログを対象としている。しかしこれは、ニュース記事等の引用を行っている発言に代表されるような1つの長い発言によって、現在のチャットルームのトピックが引きずられて、本来のトピックが抽出できないことがある。

この問題の起きる原因は、長い発言の中に存在する名詞をそのまま扱っているためだと考えられる。

### 8.2.1 問題Aに対する議論

本研究では、トピックの変化を時系列的に容易に処理できるようにするために、チャットルームのログを静的な時間間隔に分割してトピック抽出を行った。しかし、実際の会話で行われるトピックの長さは不定であるため、これをうまく反映しなければならない。

この問題に対して、トピックを動的な時間単位で抽出することを、考える必要がある。しかし、これを行うには本研究のトピックの始点と終点を決める必要がある。これは、各トピックの名詞量の変化や発言量の変化から決められると考えられる。

### 8.2.2 問題Bに対する議論

チャットルームの各時間のログにおいて、複数のトピックの抽出を実現することは実際のチャットルームの会話をうまく反映するために必要である。

この問題に対して、現在のチャットルームのログに対してだけ名詞の偏りを測るのではなく、各トピックにおいて、直前のログや現在に近い過去のログに対してもそのトピックに属する名詞の偏りを測れば良いと考える。

### 8.2.3 問題Cに対する議論

直前のログの特徴によって現在のトピックをうまく抽出できない問題に対して、トピックの変わり目、即ちトピックの始点と終点を見つける必要があると考える。

トピックの始点や終点を見つけるためには、各トピックの名詞量の変化や発言量の変化を直前のログや現在に近い過去のログに対しても測れば良いと考える。このようなトピックの変わり目がわかれば、その部分のログには直前のログの特徴を付加しないようにすれば、問題Cは解決できるだろう。

### 8.2.4 問題Dに対する議論

各ユーザの過去のログの特徴によって、そのチャットルームでは馴染みのないトピックをうまく抽出できない問題がある。

これに対して、トピックの変わり目を見つけ、その部分にだけ、各ユーザの過去のログの特徴を付加しないのであれば、トピックの推定精度は上がるかもしれない。しかし、各ユーザの過去のログの特徴を考慮することで抽出したトピックを話す土壌ができる。よって、どのトピックも正確に抽出する目的か、チャットルームの会話に参加するという目的かによって、トピックの変わり目に対して、各ユーザの過去のログの特徴を付加するかを決めれば良いだろう。

### 8.2.5 問題Eに対する議論

一つの長い発言の中に存在する大量の名詞によって、トピックが正常に抽出できない問題に対して、1つの発言の中に存在する名詞の量を対数化すると、1つの発言に引きずられずに、トピックをうまく抽出できるかもしれない。しかしこの方法は、名詞の数を減らすことになってしまうので、トピックの推定精度が落ちてしまう危険もある。また、皆がそのトピックに関わっているかを調べるために、発言しているユーザの種類も考慮すれば、より良いだろう。但し、このようなことをすると過学習になり、トピックの推定に悪い影響を及ぼしてしまう可能性がある。

# 第 9 章

# おわりに

本論文では、チャットルームの会話において、各ユーザの特徴とログの特徴を用いることで、チャットルームのトピックを抽出するシステムを提案した。

各ユーザの特徴としては、各ユーザの過去の発言履歴を現在のユーザの発言の一部とした。ログの特徴としては、現在の直前のログを現在のチャットルームのログの一部とした。また、チャットルームのログに存在する各名詞をいくつかのトピックに分類することで、トピックを抽出した。これらの特徴をマージすることで、短期間の少ない情報量のチャットのログを拡大し、文の特徴が捉え難いものの特徴をうまく捉えることで、トピックを抽出することができた。

実験結果から、本論文で提案したシステムは、信頼度が高く推定されたトピックにおいて、高い精度を示すことができた。また、直前のログの特徴を用いることで、より継続性のあるトピックを抽出できることも示すことができた。

今後は、多種多様なチャットのトピックの抽出、名詞の分類の高速化、動的なトピックの抽出、最適なログの特徴の適用等に取り組んで行く必要がある。

# 謝辞

本研究を遂行するにあたっては、いろいろな方々にお世話になりました。

まず、指導教員の村山隆彦先生には日頃から熱心なご指導、そしてご鞭撻を賜わりました。また、ご多忙中にもかかわらず論文の草稿を丁寧に読んで下さり、大変貴重なご助言をいただきました。ここに厚く御礼申し上げます。

また、研究を進めるにあたり、多田好克先生、小宮常康先生、水野修先生、佐藤喬先生にも貴重なご助言を頂きましたことを、ここに深く感謝します。

そして、本研究が行なえたことは、研究方針や方法論について議論をし、共に研究生活をおくってきた多田研、小宮研、水野研の学生諸氏のおかげでもあります。最後に、これらの皆さんに感謝いたします。

# 参考文献

[1] Twitter, http://twitter.com/.

[2] 野美山 浩, 新聞記事データベースからの話題の抽出, 情報処理学会第 50 回全国大会, Vol.4, pp.45-46, 1994.

[3] 関口 裕一郎, 川島 晴美, 奥田 英範, 奥 雅博, ブログ発信者の特徴を利用した話題抽出手法, 日本データベース学会 Letters Vol.5, No.1, pp.9-12, 2006.

[4] 石井 恵, 中渡瀬 秀一, 富田 準二, 名詞句と単語の勢いを用いた話題抽出手法の提案, 第 160 回情報処理学会自然言語処理研究会報告, pp.79-84, 2004.

[5] 石田基広, R によるテキストマイニング入門, 森北出版株式会社, pp.45, 2008.

[6] The Perl Programming Language, http://www.perl.org/.

[7] The R Project for Statistical Computing, http://www.r-project.org/.

[8] 石田基広, R によるテキストマイニング入門, 森北出版株式会社, pp.45-82, 2008.

[9] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://mecab.sourceforge.net/.

[10] 独立行政方針 国立国語研究所：分類語彙表ー-増補改訂版データベース, 2004.

[11] Yahoo!デベロッパーネットワークの Yahoo!検索 Web API, http://developer.yahoo.co.jp/webapi/search/.

[12] シンプソン係数とは (Simpson's Coefficient) シンプソンけいすう:–IT 用語辞典バイナリ:, http://www.sophia-it.com/content/シンプソン係数.

# 付録A

# チャットログとニュース記事の品詞構成図

チャットログの品詞構成（%）(275623要素)

| | | | |
|---|---|---|---|
| センリョウ | 感動詞-フィラー | 接尾辞-名詞的-一般 | 接尾辞-名詞的-助数詞 |
| 0.00 | 1.55 | 1.61 | 0.63 |
| 感動詞-一般 | 記号-一般 | 接尾辞-名詞的-副詞可能 | 代名詞 |
| 2.28 | 0.29 | 0.08 | 1.43 |
| 記号-文字 | 空白 | 動詞-一般 | 動詞-非自立可能 |
| 15.49 | 3.21 | 3.86 | 3.57 |
| 形状詞-タリ | 形状詞-一般 | 副詞 | 補助記号-一般 |
| 0.02 | 0.27 | 1.73 | 5.78 |
| 形状詞-助動詞語幹 | 形容詞-一般 | 補助記号-括弧開 | 補助記号-括弧閉 |
| 0.18 | 0.94 | 0.91 | 0.54 |
| 形容詞-非自立可能 | 助詞-格助詞 | 補助記号-句点 | 補助記号-読点 |
| 0.90 | 6.09 | 3.86 | 1.14 |
| 助詞-係助詞 | 助詞-終助詞 | 未知語 | 名詞-固有名詞-一般 |
| 2.49 | 3.70 | 0.60 | 0.13 |
| 助詞-準体助詞 | 助詞-接続助詞 | 名詞-固有名詞-人名-一般 | 名詞-固有名詞-人名-姓 |
| 0.98 | 2.29 | 0.16 | 0.21 |
| 助詞-副助詞 | 助動詞 | 名詞-固有名詞-人名-名 | 名詞-固有名詞-組織名 |
| 1.58 | 8.01 | 0.24 | 0.12 |
| 接続詞 | 接頭辞 | 名詞-固有名詞-地名-一般 | 名詞-固有名詞-地名-国 |
| 0.33 | 0.80 | 0.32 | 0.29 |
| 接尾辞-形状詞的 | 接尾辞-形容詞的 | 名詞-助動詞語幹 | 名詞-数詞 |
| 0.14 | 0.08 | 0.00 | 2.28 |
| 接尾辞-動詞的 | 接尾辞-名詞的-サ変可能 | 名詞-普通名詞-サ変可能 | 名詞-普通名詞-サ変形状詞可能 |
| 0.06 | 0.02 | 4.20 | 0.10 |
| 接尾辞-名詞的-一般 | 接尾辞-名詞的-助数詞 | 名詞-普通名詞-一般 | 名詞-普通名詞-形状詞可能 |
| 1.61 | 0.63 | 11.83 | 0.87 |
| | | 名詞-普通名詞-副詞可能 | 連体詞 |
| | | 1.40 | 0.41 |

図 A.1: チャットログの品詞構成

ニュース記事の品詞構成(%)(36329要素)

| | | | | | |
|---|---|---|---|---|---|
| 感動詞-フィラー | 感動詞-一般 | 接尾辞-名詞的-副詞可能 | 代名詞 | | |
| 0.06 | 0.02 | 0.18 | 0.39 | | |
| 記号-一般 | 記号-文字 | 動詞-一般 | 動詞-非自立可能 | | |
| 0.02 | 0.15 | 5.11 | 5.27 | | |
| 空白 | 形状詞-一般 | 副詞 | 補助記号-一般 | | |
| 1.28 | 0.38 | 0.76 | 0.36 | | |
| 形状詞-助動詞語幹 | 形容詞-一般 | 補助記号-括弧開 | 補助記号-括弧閉 | | |
| 0.22 | 0.66 | 1.30 | 1.29 | | |
| 形容詞-非自立可能 | 助詞-格助詞 | 補助記号-句点 | 補助記号-読点 | | |
| 0.40 | 17.02 | 3.41 | 4.74 | | |
| 助詞-係助詞 | 助詞-終助詞 | 名詞-固有名詞-一般 | 名詞-固有名詞-人名-一般 | | |
| 3.53 | 0.21 | 0.06 | 0.13 | | |
| 助詞-準体助詞 | 助詞-接続助詞 | 名詞-固有名詞-人名-姓 | 名詞-固有名詞-人名-名 | | |
| 0.47 | 2.76 | 0.51 | 0.35 | | |
| 助詞-副助詞 | 助動詞 | 名詞-固有名詞-組織名 | 名詞-固有名詞-地名-一般 | | |
| 1.38 | 5.88 | 0.31 | 0.86 | | |
| 接続詞 | 接頭辞 | 名詞-固有名詞-地名-国 | 名詞-数詞 | | |
| 0.22 | 0.98 | 0.54 | 5.74 | | |
| 接尾辞-形状詞的 | 接尾辞-形容詞的 | 名詞-普通名詞-サ変可能 | 名詞-普通名詞-サ変形状詞可能 | | |
| 0.14 | 0.03 | 7.40 | 0.10 | | |
| 接尾辞-動詞的 | 接尾辞-名詞的-サ変可能 | 名詞-普通名詞-一般 | 名詞-普通名詞-形状詞可能 | | |
| 0.01 | 0.12 | 15.87 | 0.78 | | |
| 接尾辞-名詞的-一般 | 接尾辞-名詞的-助数詞 | 名詞-普通名詞-副詞可能 | 連体詞 | | |
| 2.92 | 2.78 | 2.37 | 0.52 | | |

図 A.2: ニュース記事の品詞構成

# 付録B

# トピック推定結果

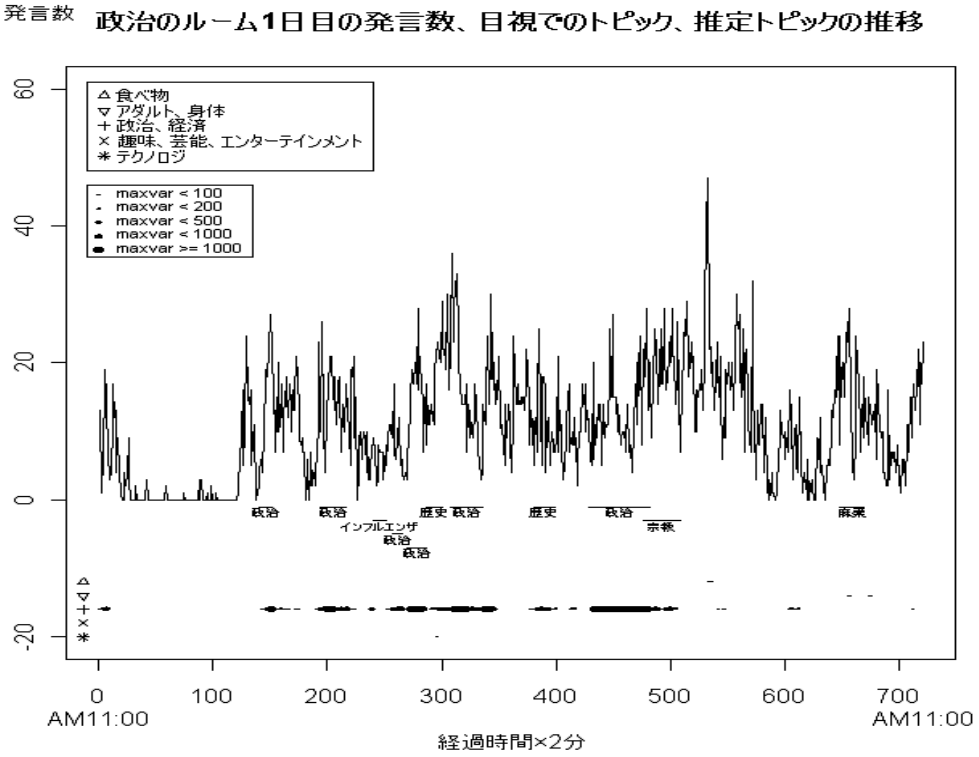


図 B.1: 政治の 1 日目の発言数、目視でのトピック、推定トピックの推移

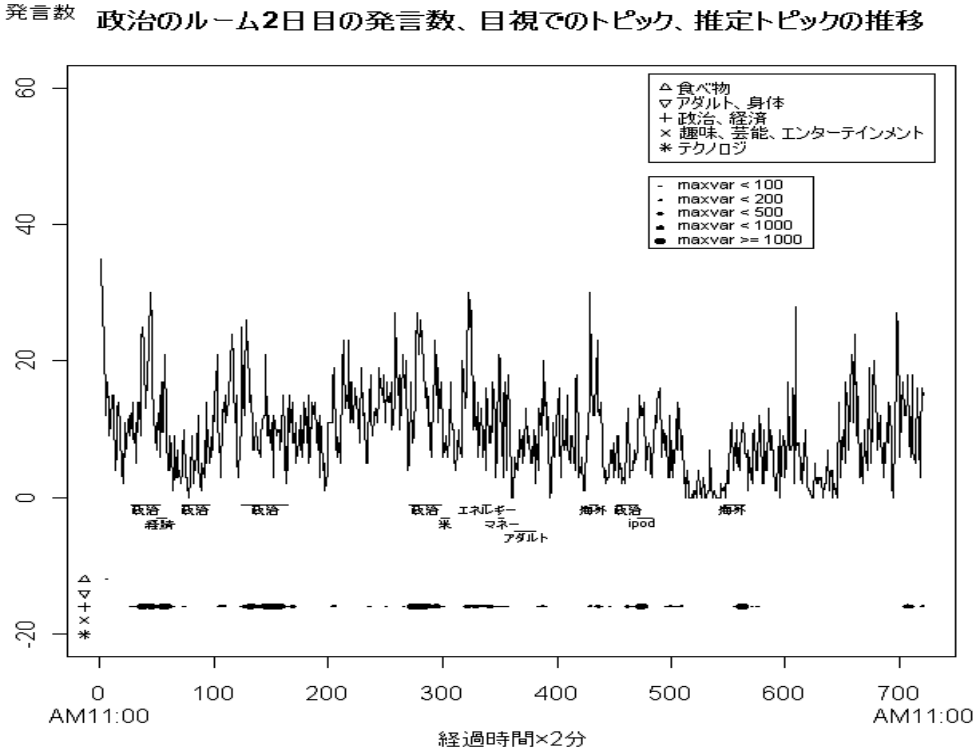


図 B.2: 政治の 2 日目の発言数、目視でのトピック、推定トピックの推移

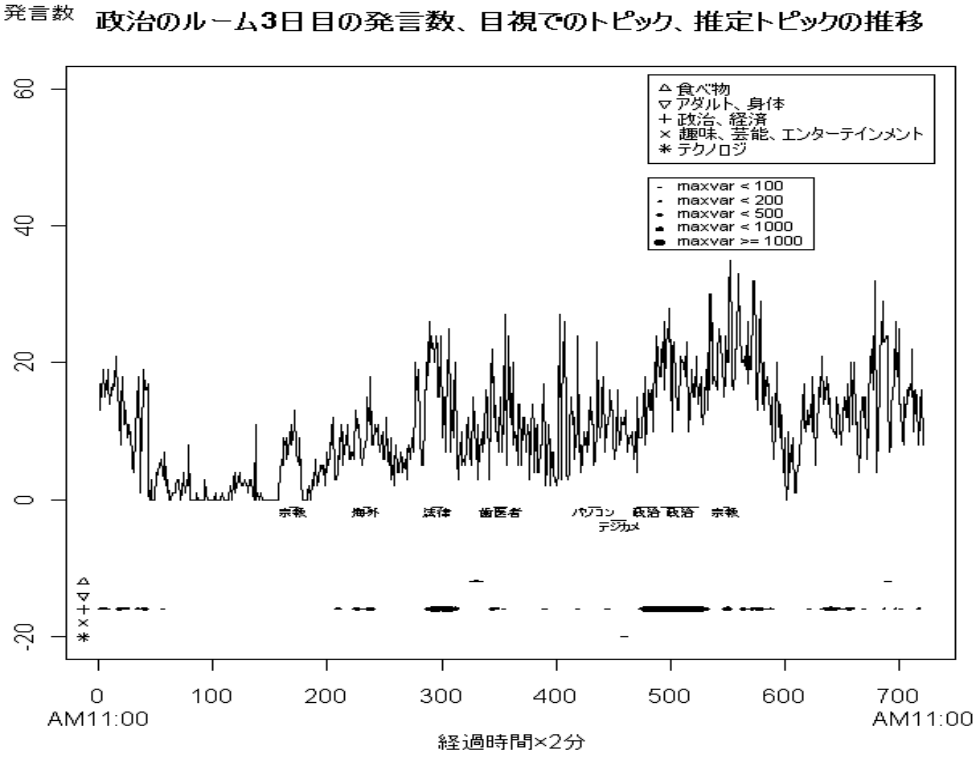


図 B.3: 政治の 3 日目の発言数、目視でのトピック、推定トピックの推移

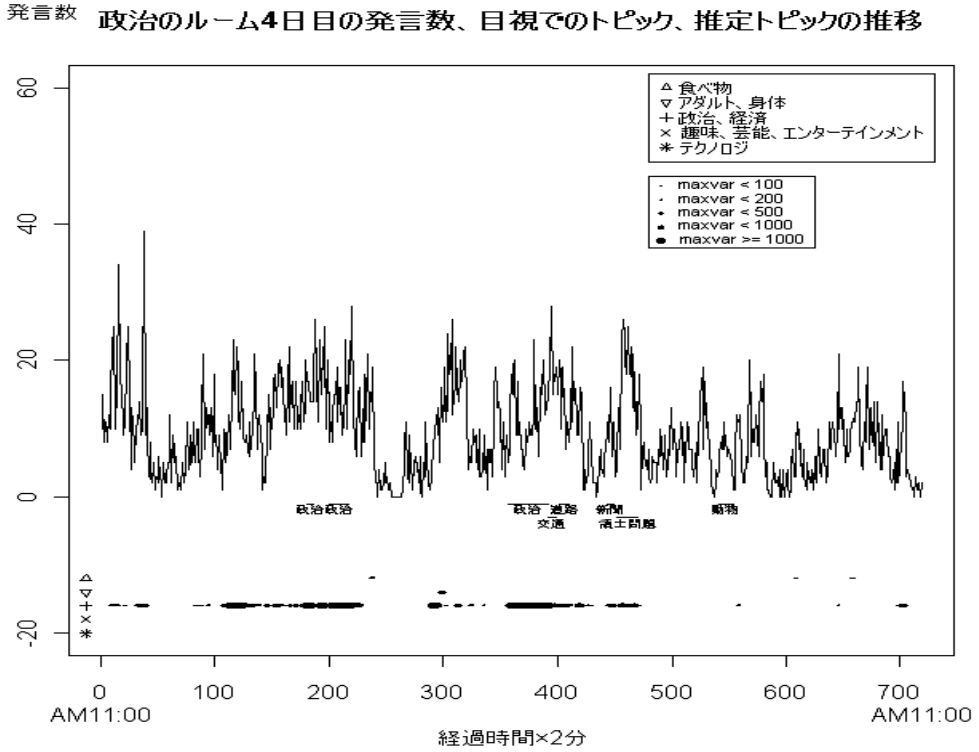


図 B.4: 政治の 4 日目の発言数、目視でのトピック、推定トピックの推移

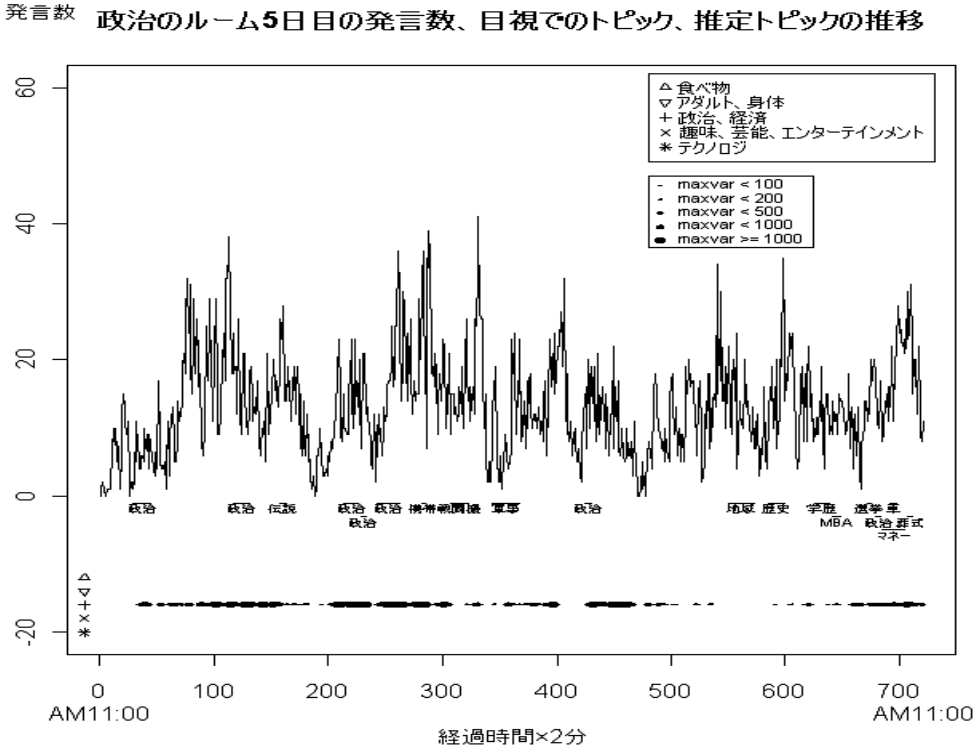


図 B.5: 政治の 5 日目の発言数、目視でのトピック、推定トピックの推移

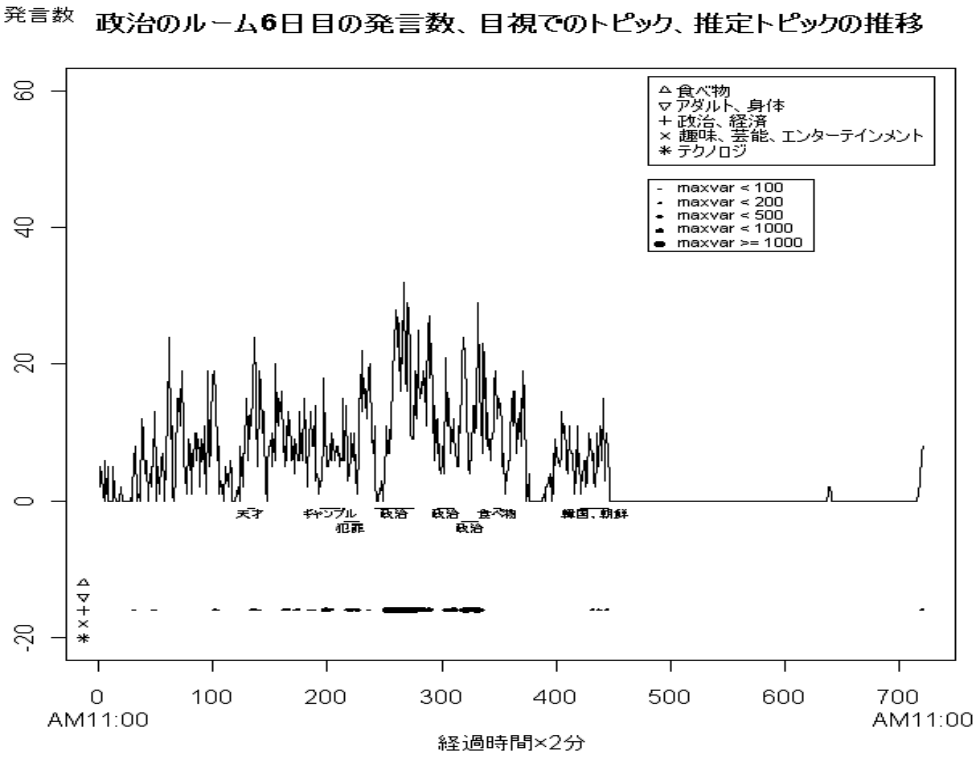


図 B.6: 政治の 6 日目の発言数、目視でのトピック、推定トピックの推移

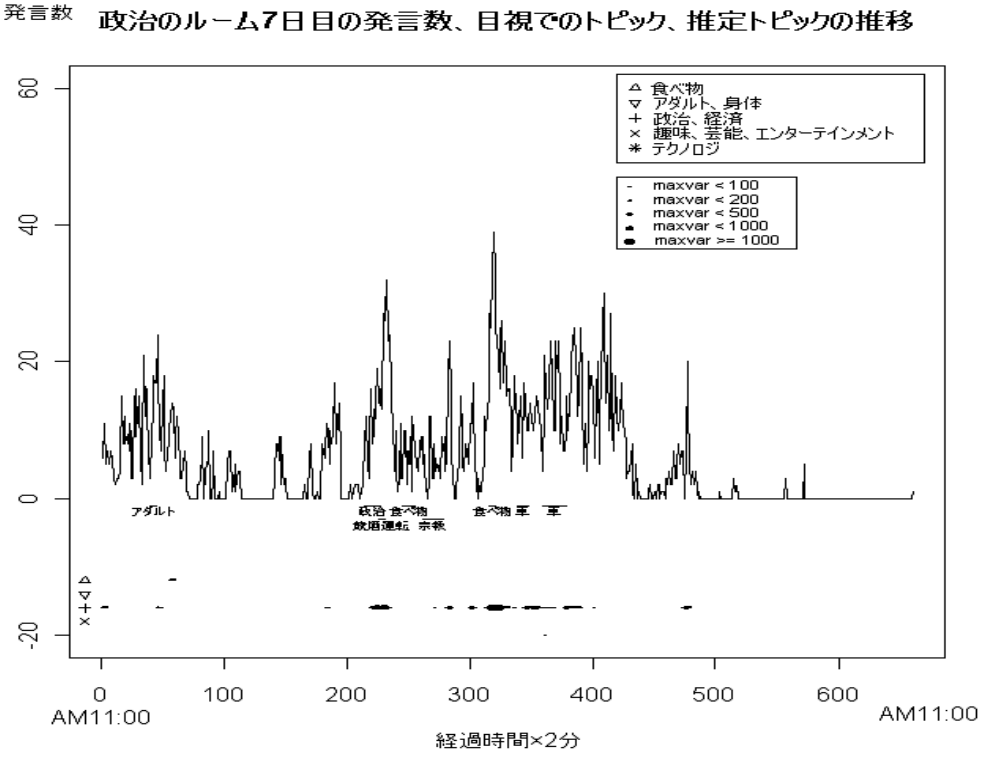


図 B.7: 政治の 7 日目の発言数、目視でのトピック、推定トピックの推移

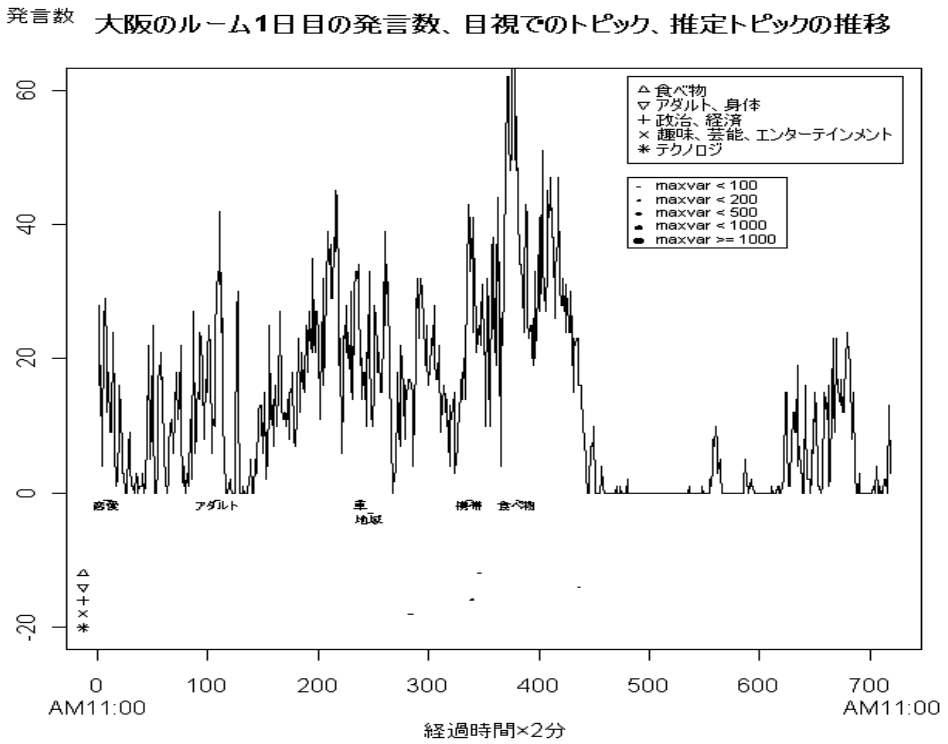


図 B.8: 大阪の 1 日目の発言数、目視でのトピック、推定トピックの推移

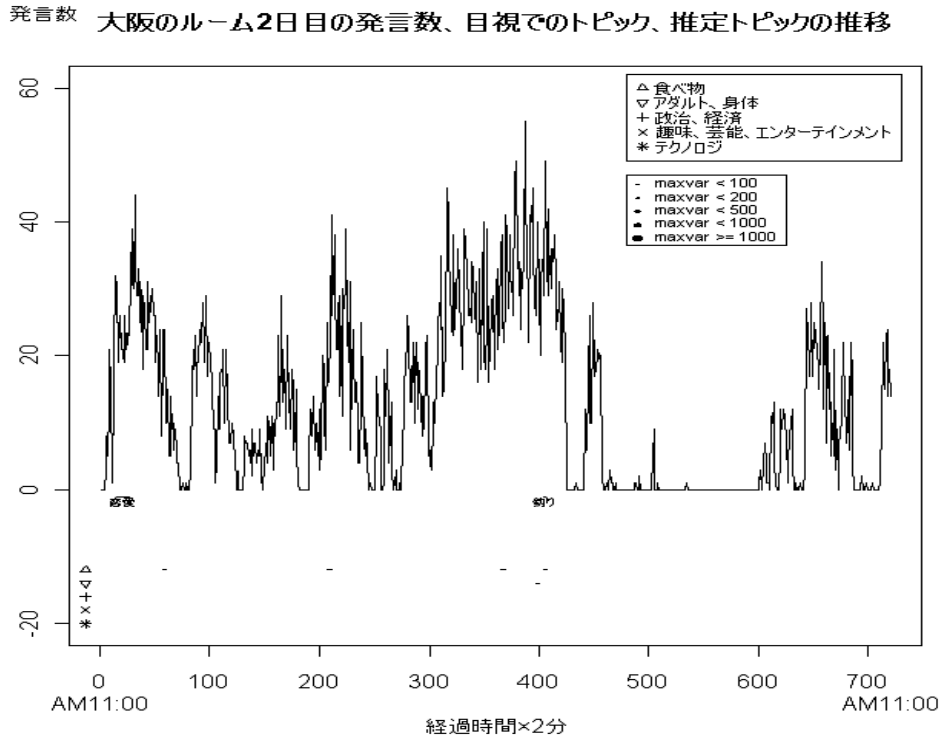


図 B.9: 大阪の 2 日目の発言数、目視でのトピック、推定トピックの推移

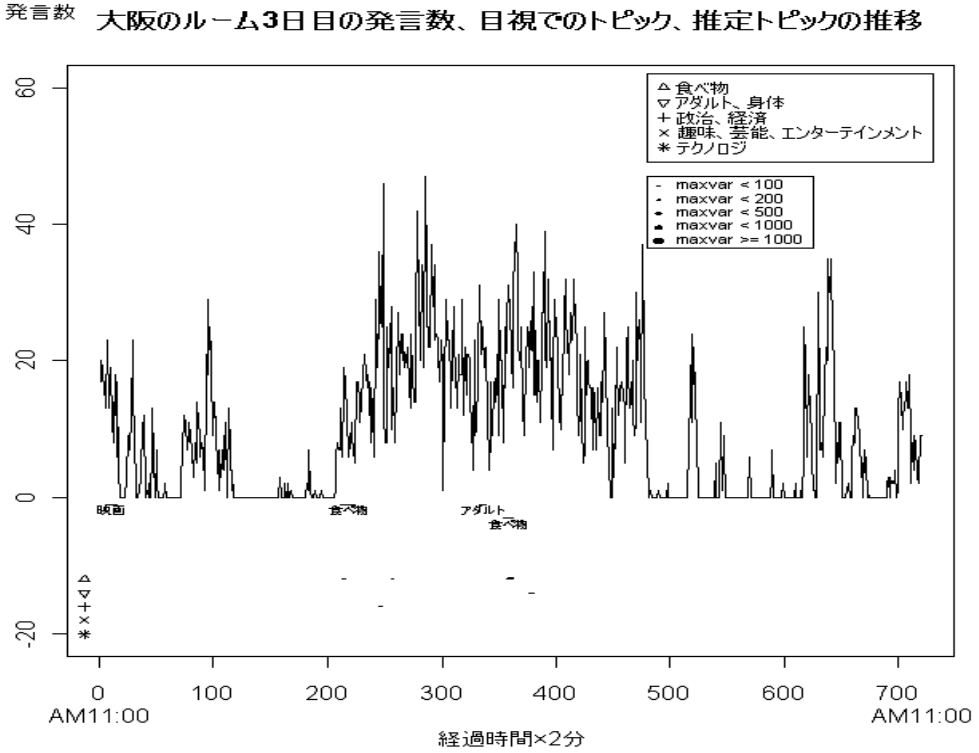


図 B.10: 大阪の 3 日目の発言数、目視でのトピック、推定トピックの推移

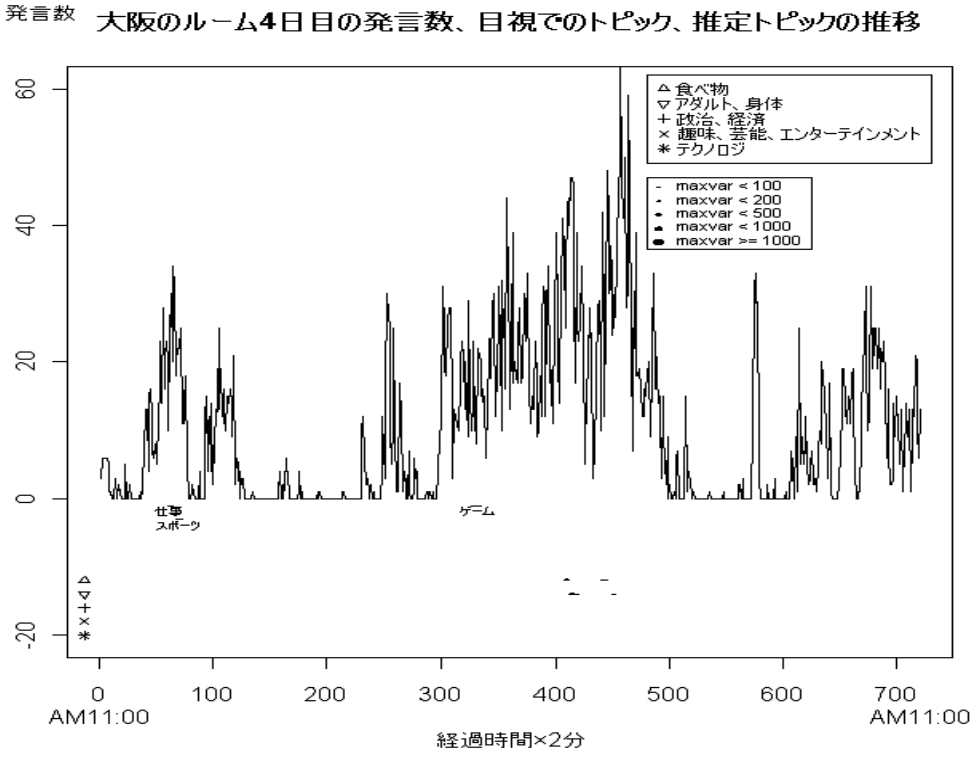


図 B.11: 大阪の 4 日目の発言数、目視でのトピック、推定トピックの推移

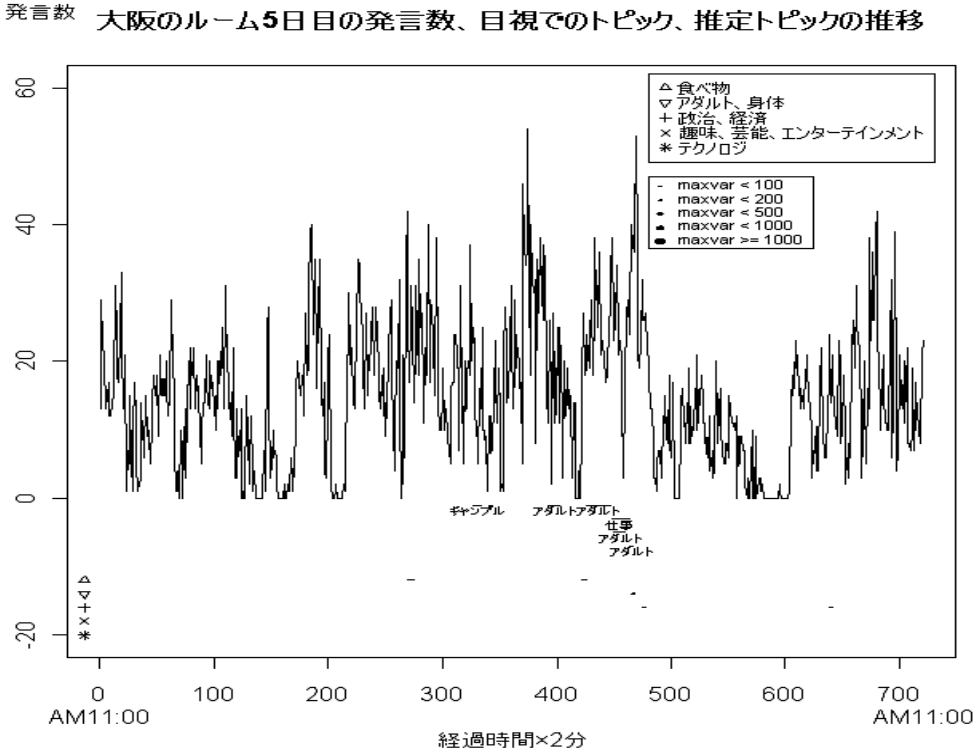


図 B.12: 大阪の 5 日目の発言数、目視でのトピック、推定トピックの推移

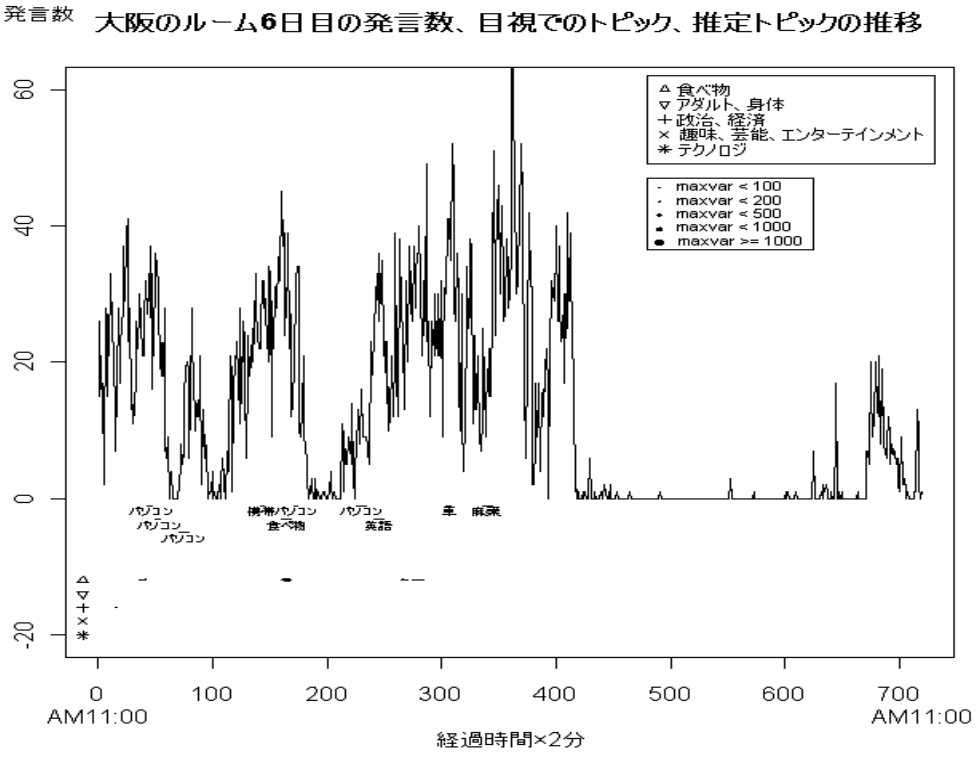


図 B.13: 大阪の 6 日目の発言数、目視でのトピック、推定トピックの推移

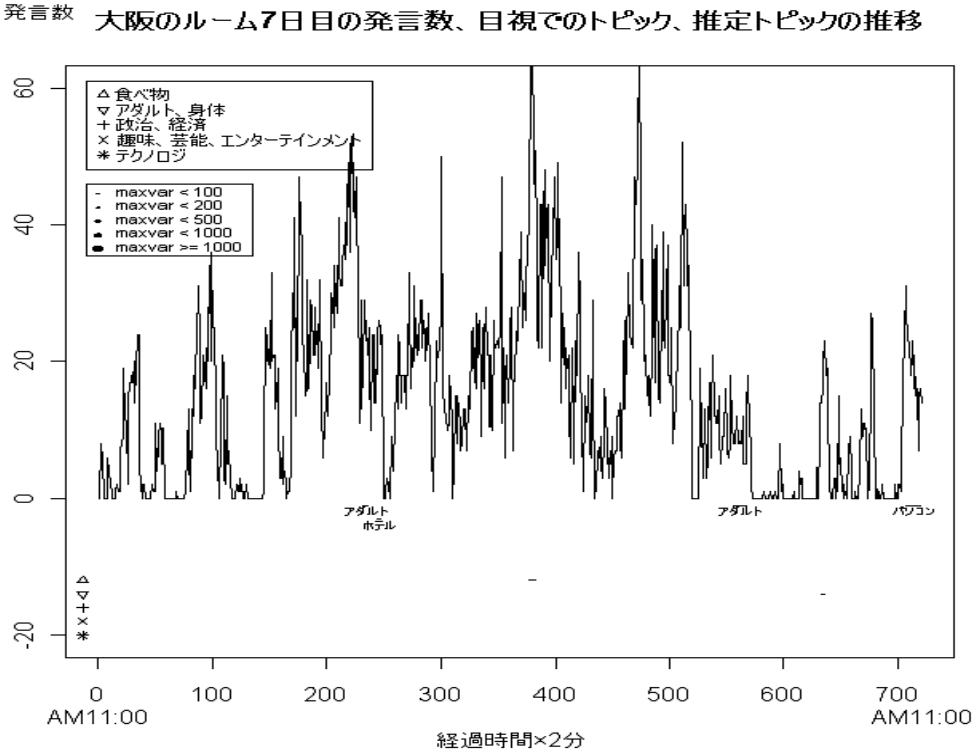


図 B.14: 大阪の 7 日目の発言数、目視でのトピック、推定トピックの推移

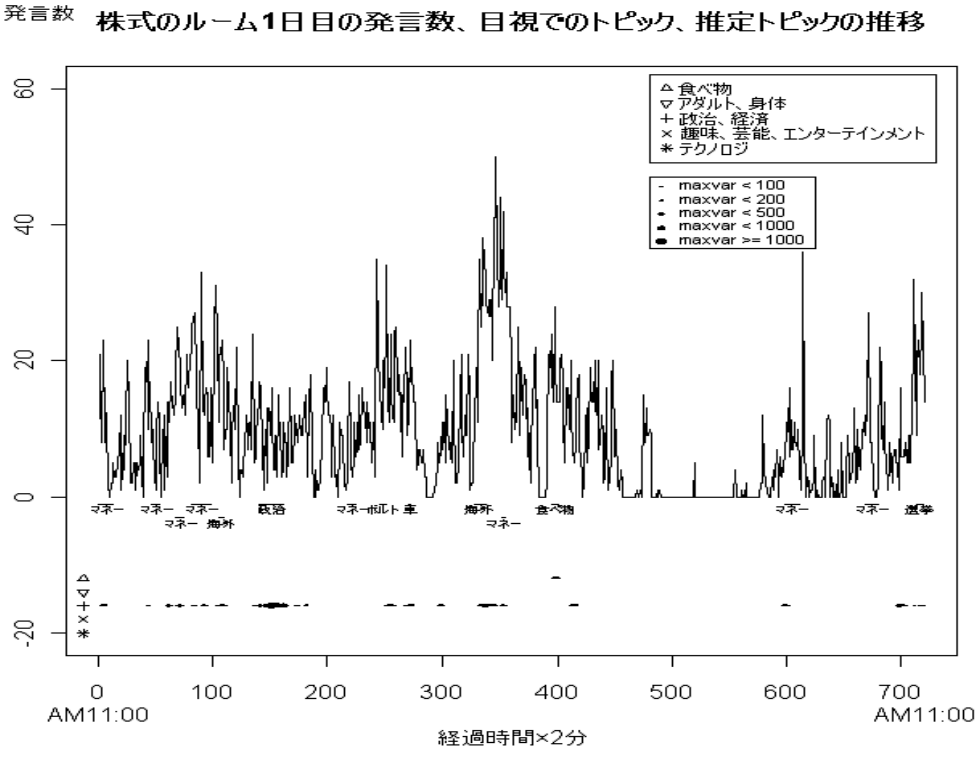


図 B.15: 株式の 1 日目の発言数、目視でのトピック、推定トピックの推移

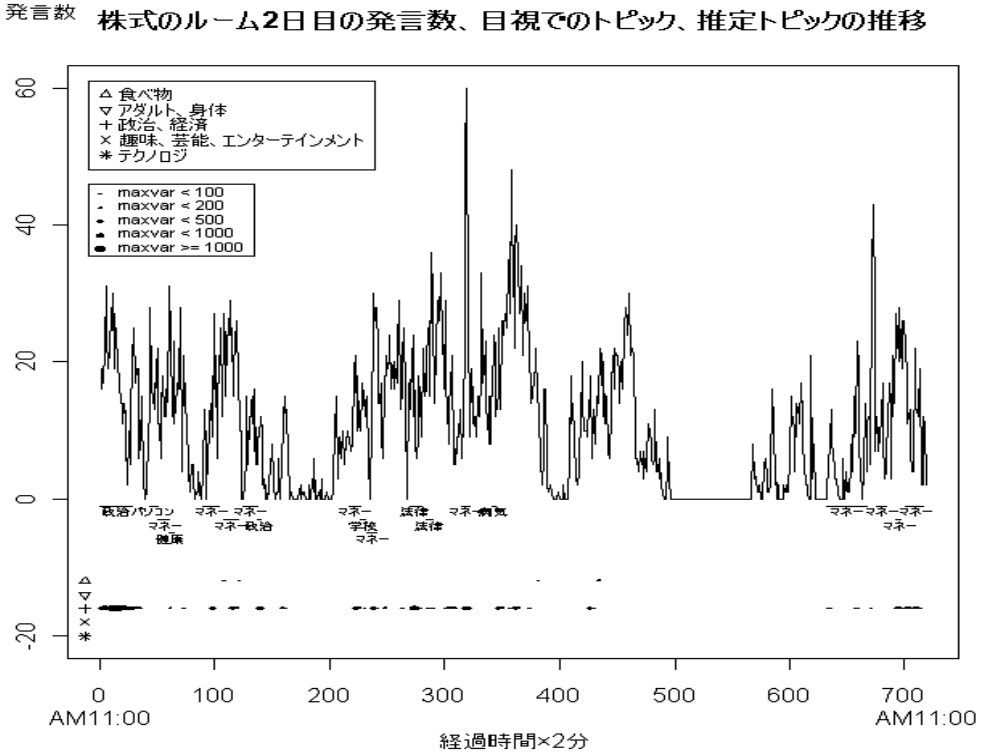


図 B.16: 株式の 2 日目の発言数、目視でのトピック、推定トピックの推移

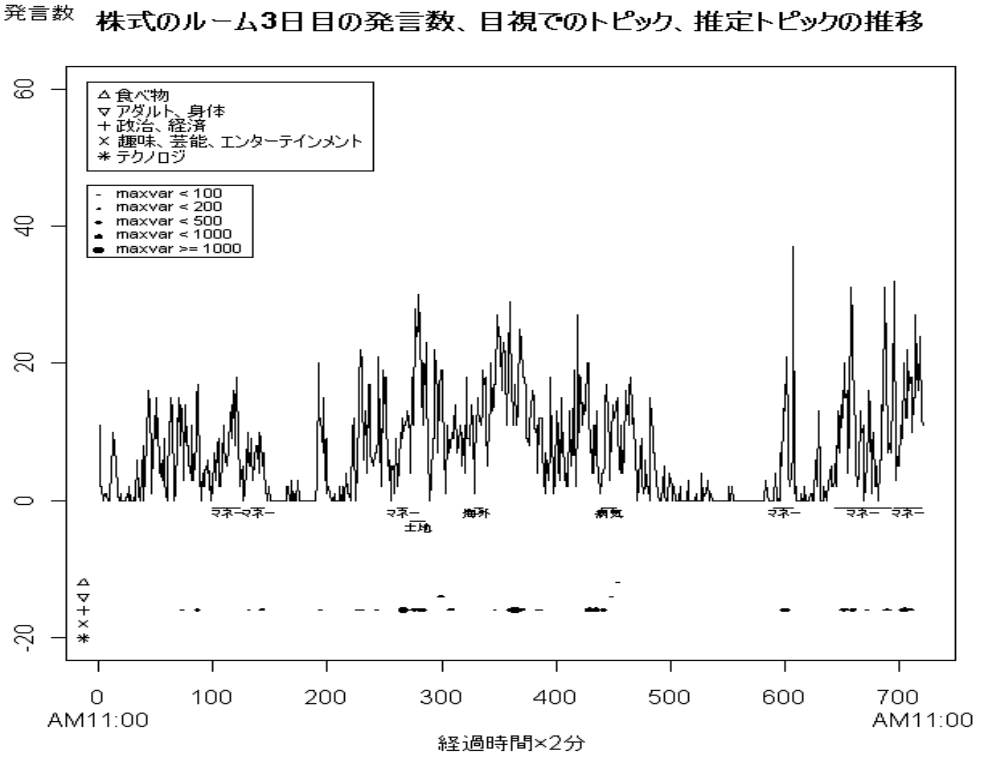


図 B.17: 株式の 3 日目の発言数、目視でのトピック、推定トピックの推移

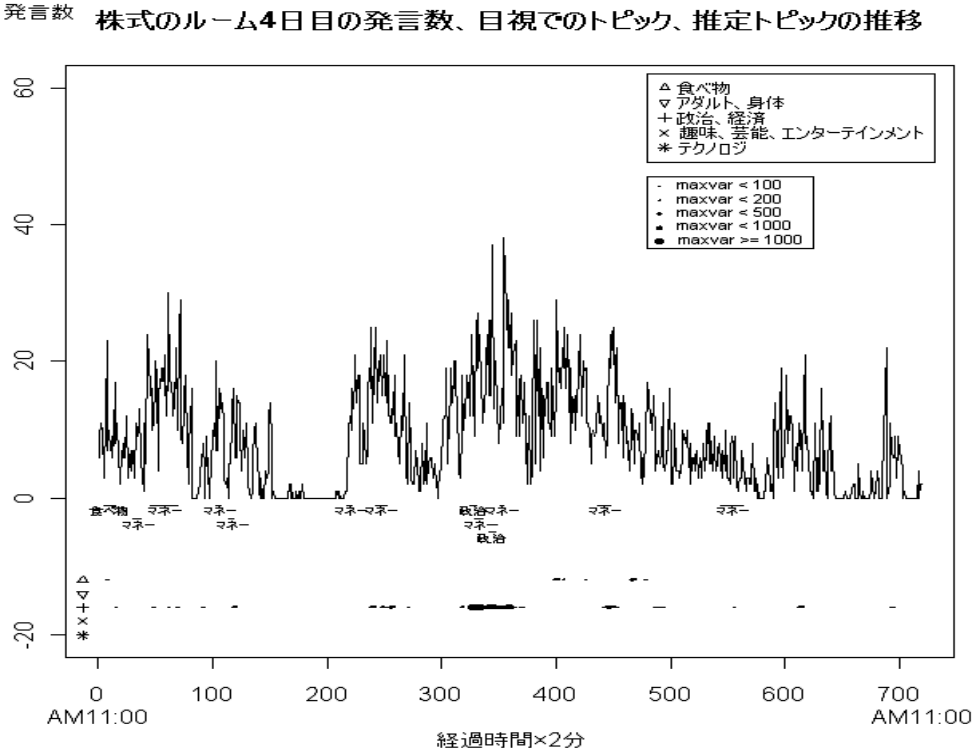


図 B.18: 株式の 4 日目の発言数、目視でのトピック、推定トピックの推移

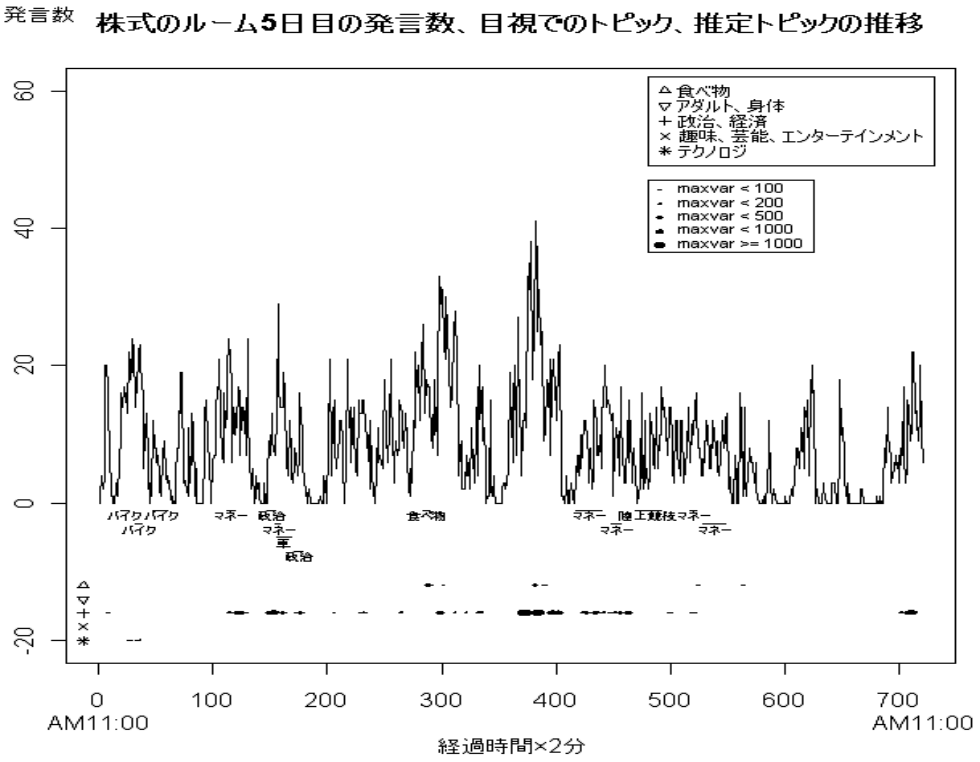


図 B.19: 株式の 5 日目の発言数、目視でのトピック、推定トピックの推移

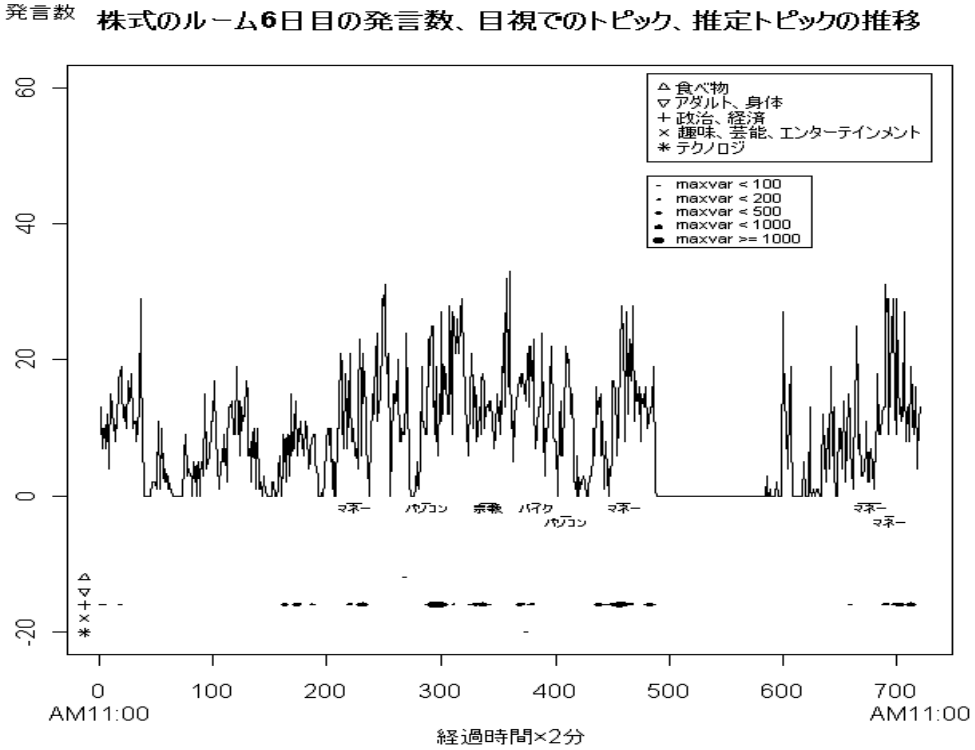


図 B.20: 株式の 6 日目の発言数、目視でのトピック、推定トピックの推移

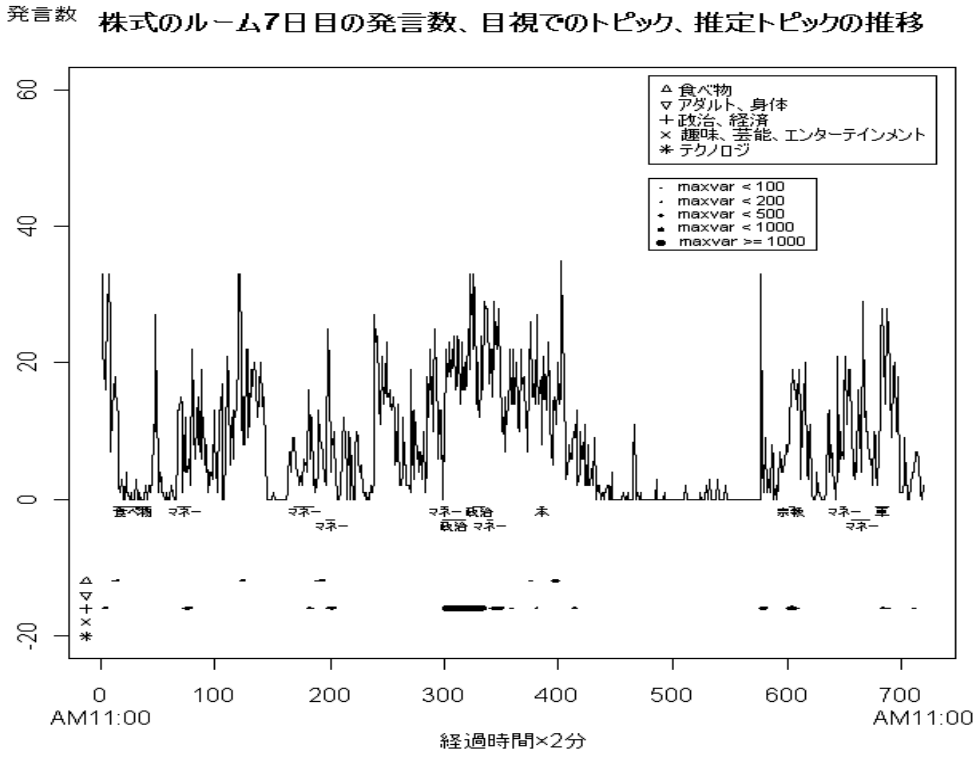


図 B.21: 株式の 7 日目の発言数、目視でのトピック、推定トピックの推移

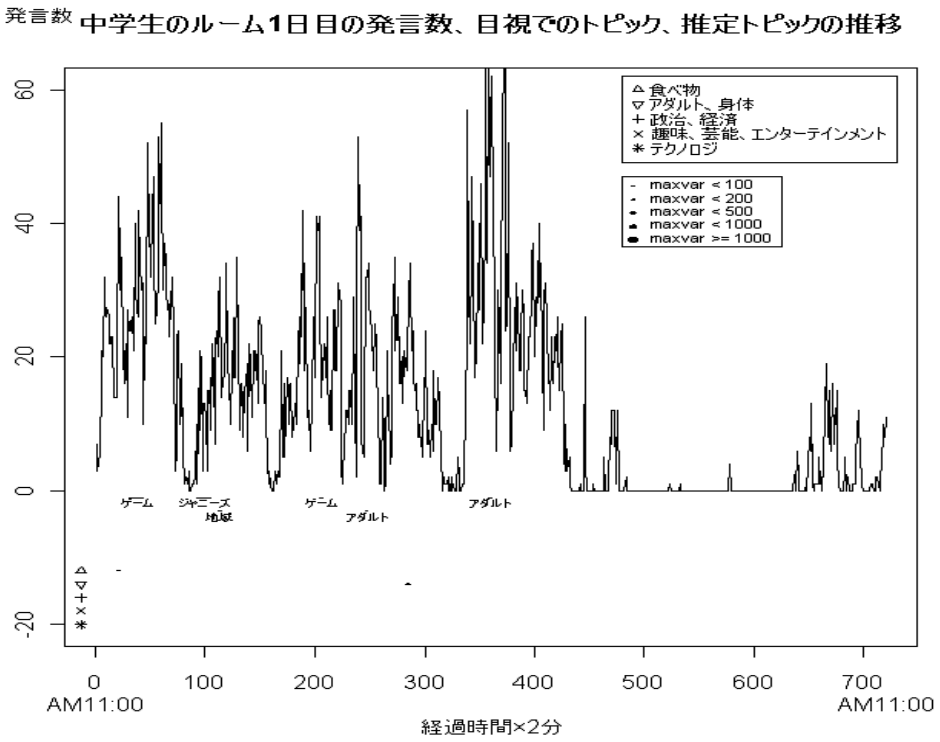


図 B.22: 中学生の 1 日目の発言数、目視でのトピック、推定トピックの推移

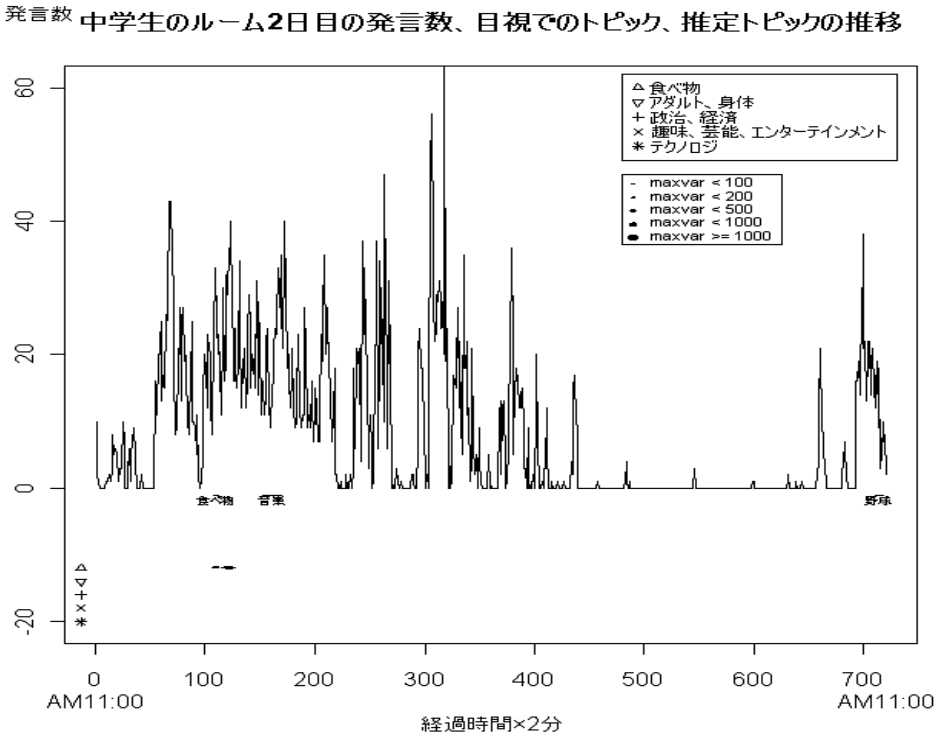


図 B.23: 中学生の 2 日目の発言数、目視でのトピック、推定トピックの推移

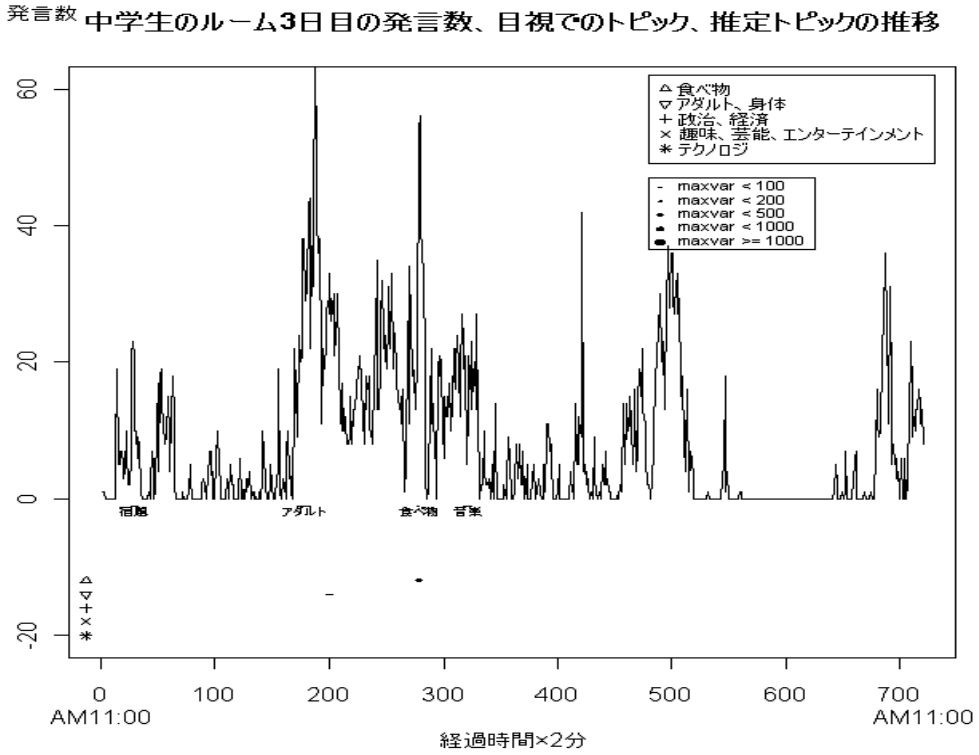


図 B.24: 中学生の 3 日目の発言数、目視でのトピック、推定トピックの推移

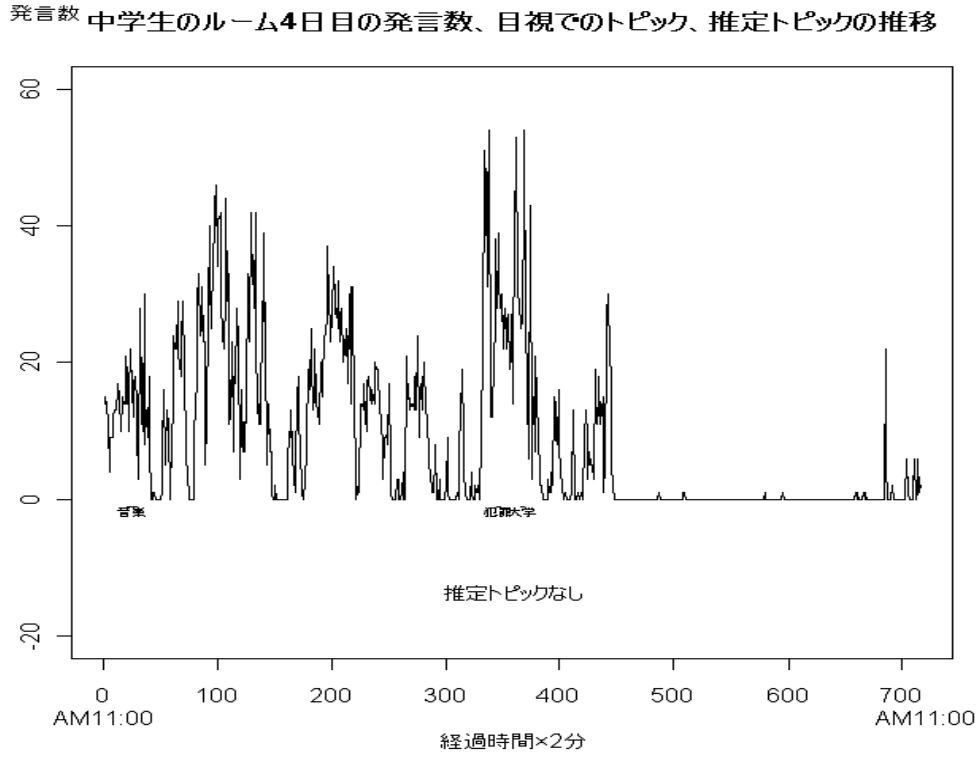


図 B.25: 中学生の 4 日目の発言数、目視でのトピック、推定トピックの推移

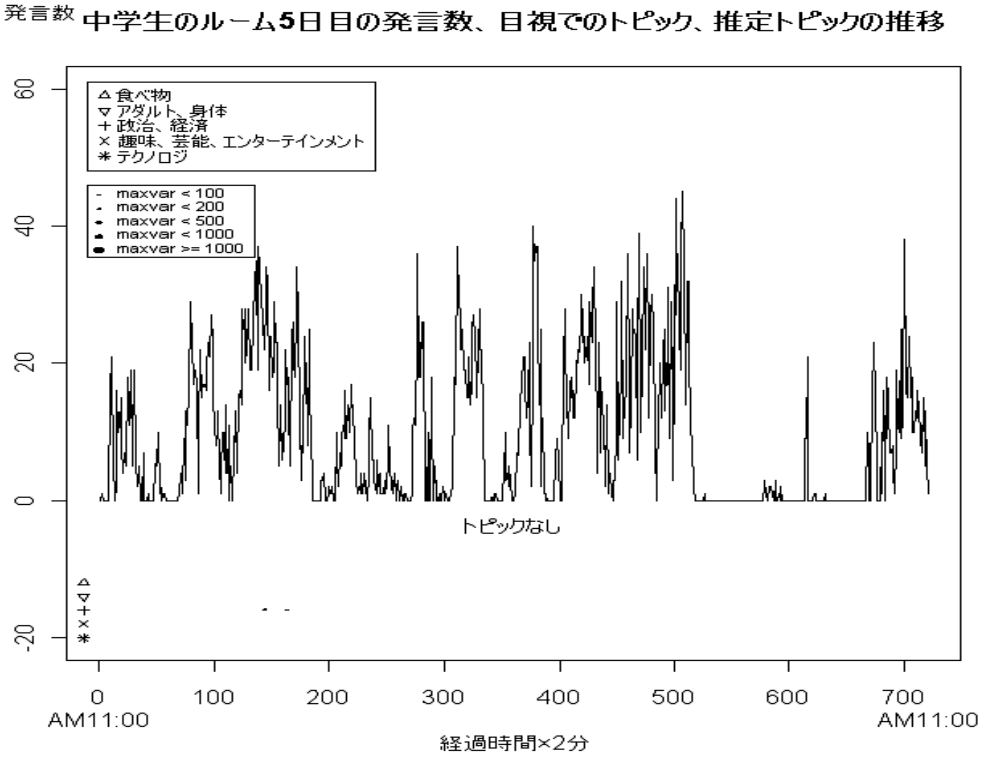


図 B.26: 中学生の 5 日目の発言数、目視でのトピック、推定トピックの推移

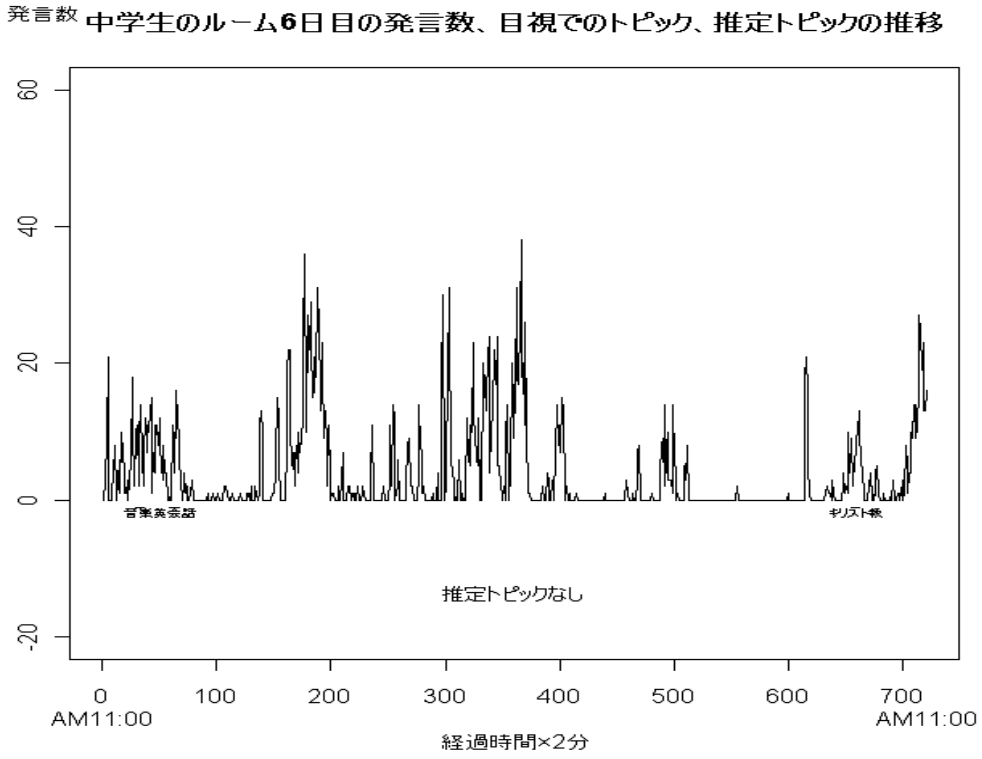


図 B.27: 中学生の 6 日目の発言数、目視でのトピック、推定トピックの推移

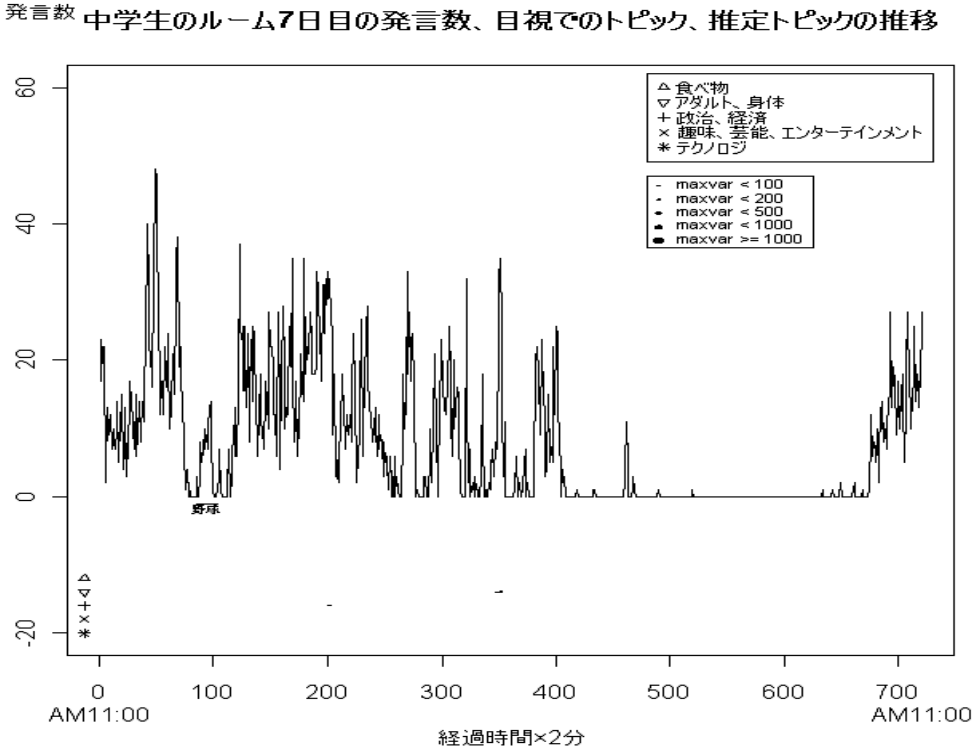


図 B.28: 中学生の 7 日目の発言数、目視でのトピック、推定トピックの推移

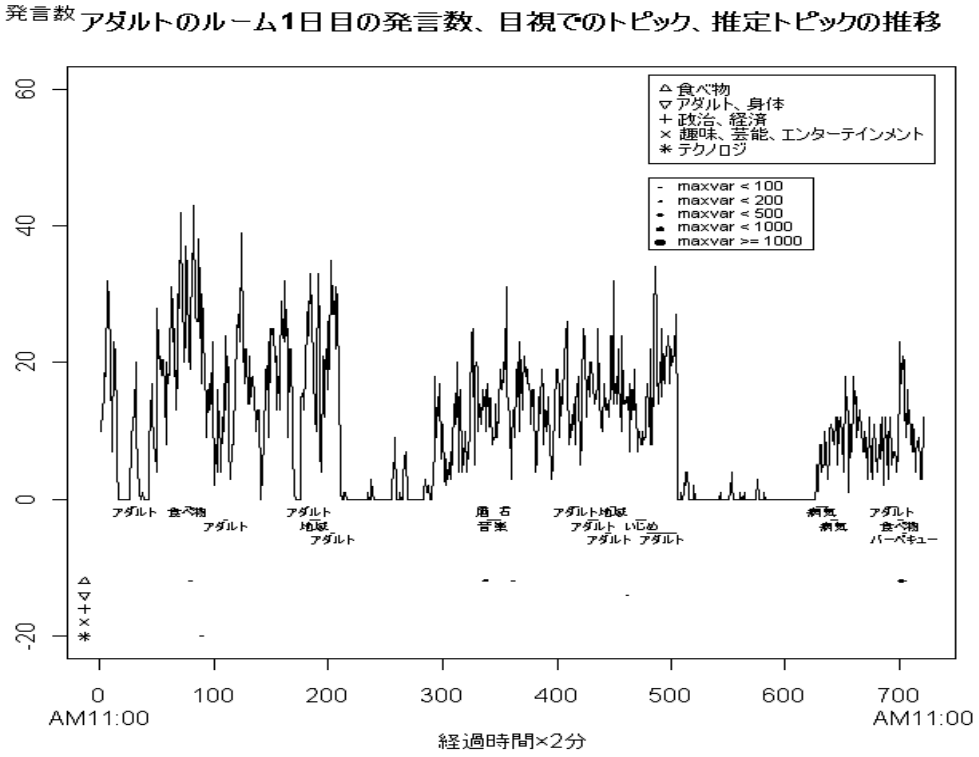


図 B.29: アダルトの 1 日目の発言数、目視でのトピック、推定トピックの推移

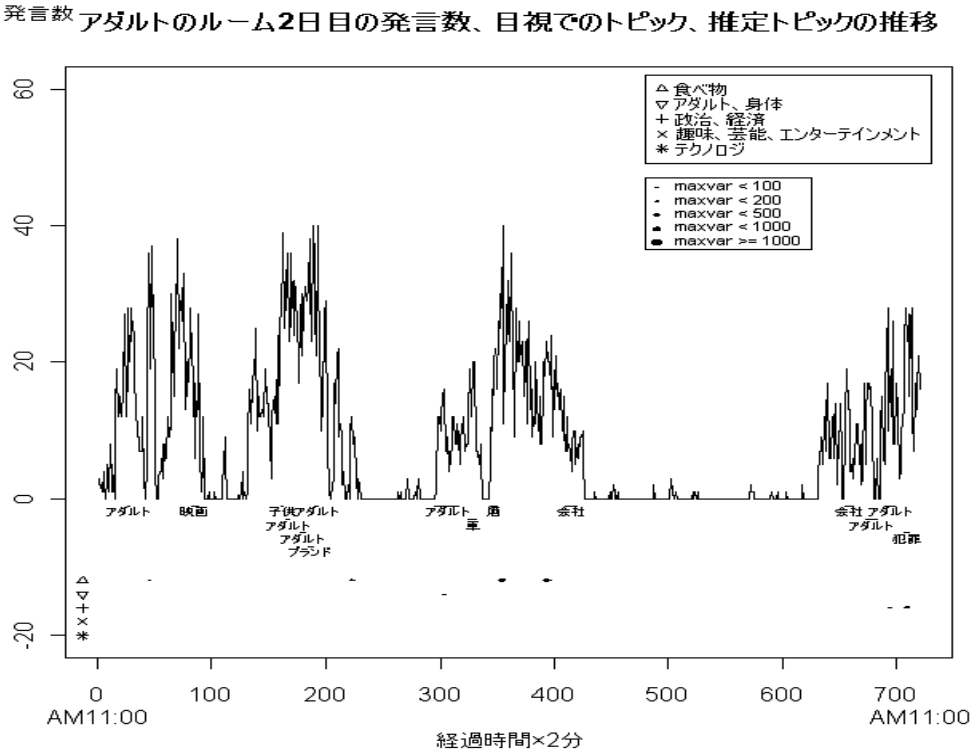


図 B.30: アダルトの 2 日目の発言数、目視でのトピック、推定トピックの推移

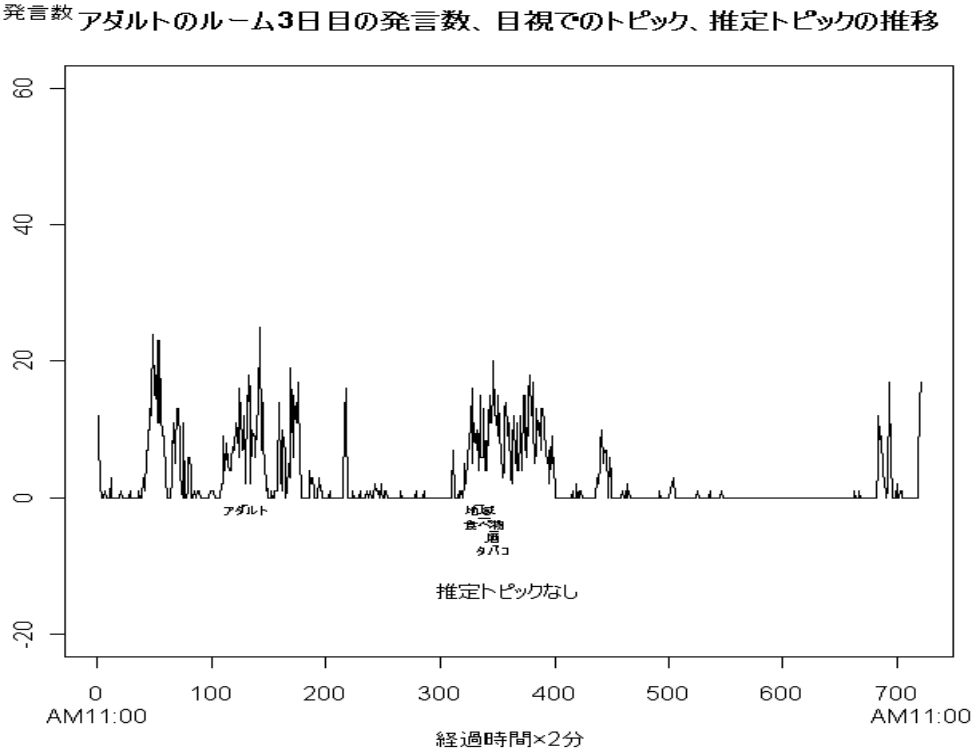


図 B.31: アダルトの 3 日目の発言数、目視でのトピック、推定トピックの推移

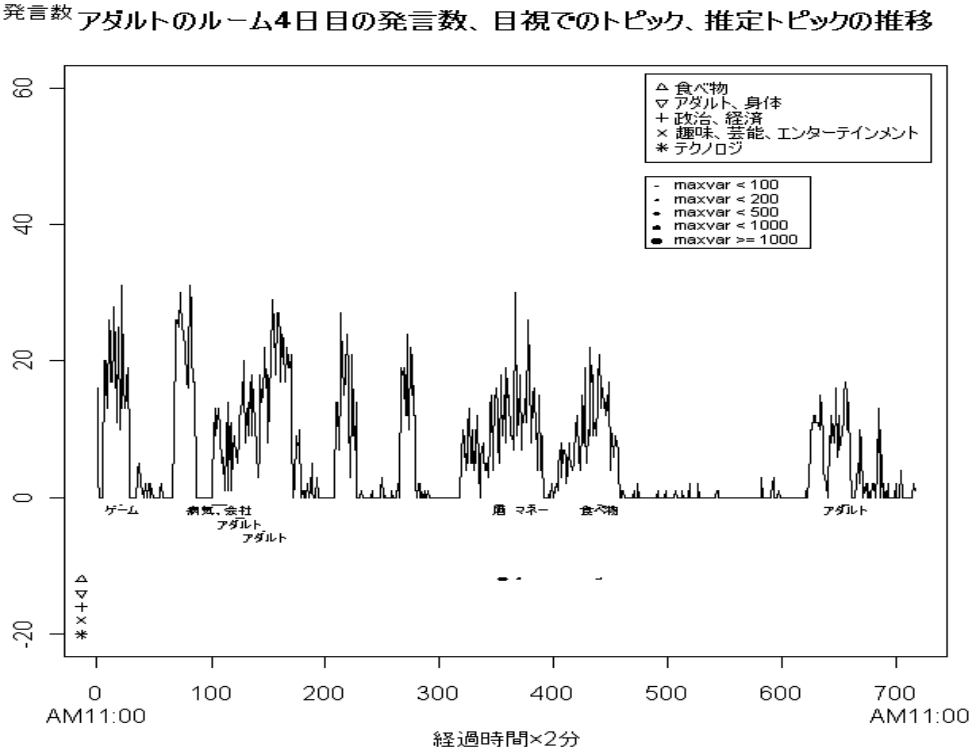


図 B.32: アダルトの 4 日目の発言数、目視でのトピック、推定トピックの推移

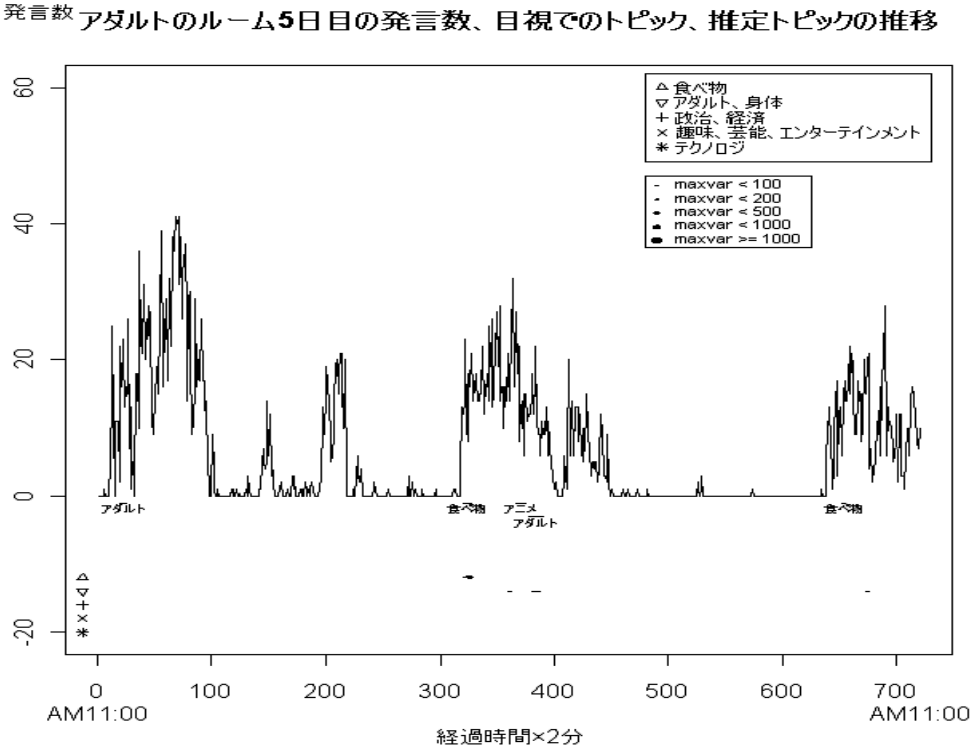


図 B.33: アダルトの 5 日目の発言数、目視でのトピック、推定トピックの推移

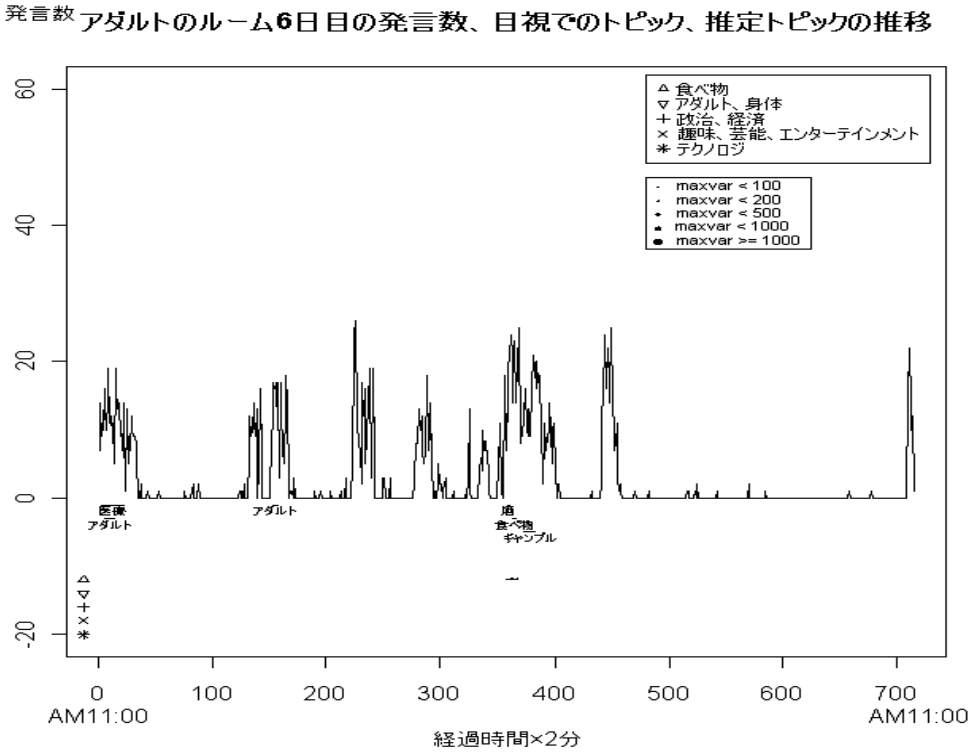


図 B.34: アダルトの 6 日目の発言数、目視でのトピック、推定トピックの推移

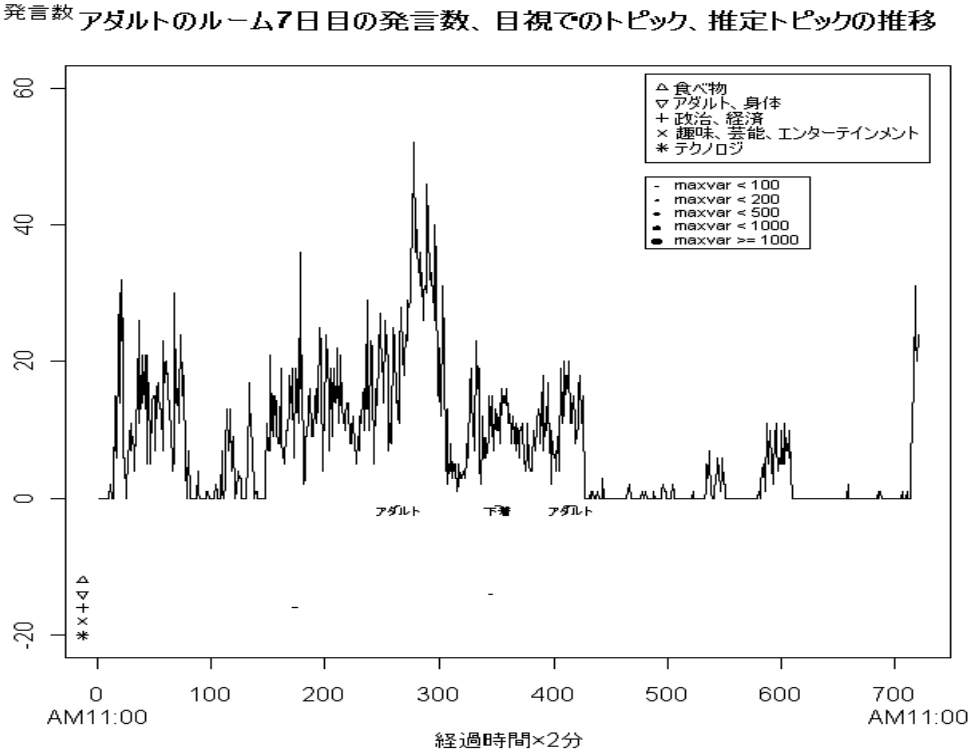


図 B.35: アダルトの 7 日目の発言数、目視でのトピック、推定トピックの推移

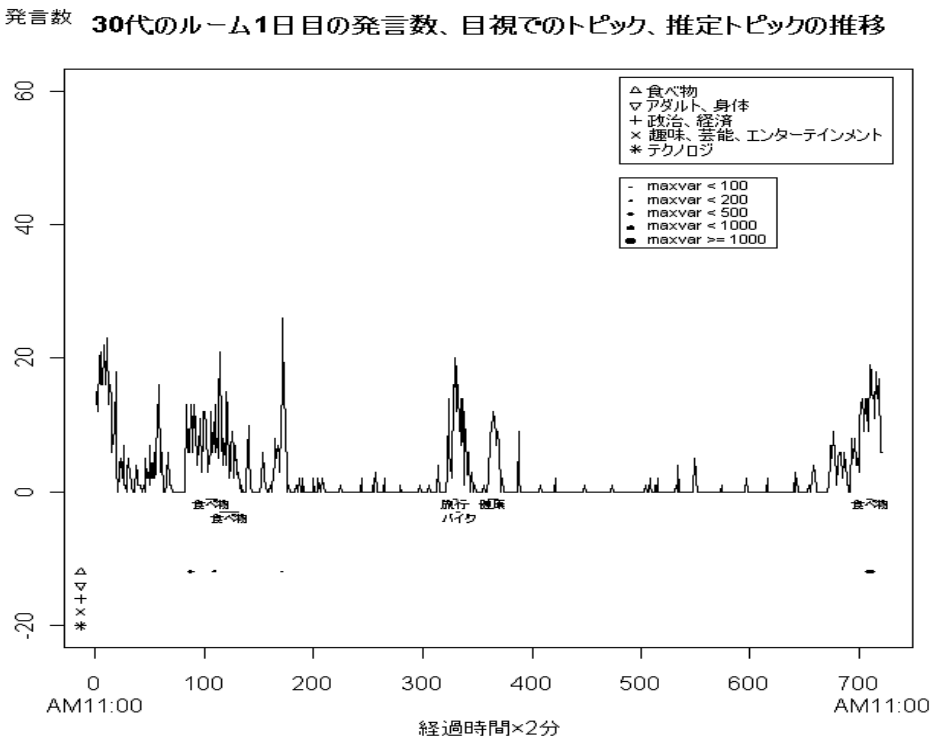


図 B.36: 30 代の 1 日目の発言数、目視でのトピック、推定トピックの推移

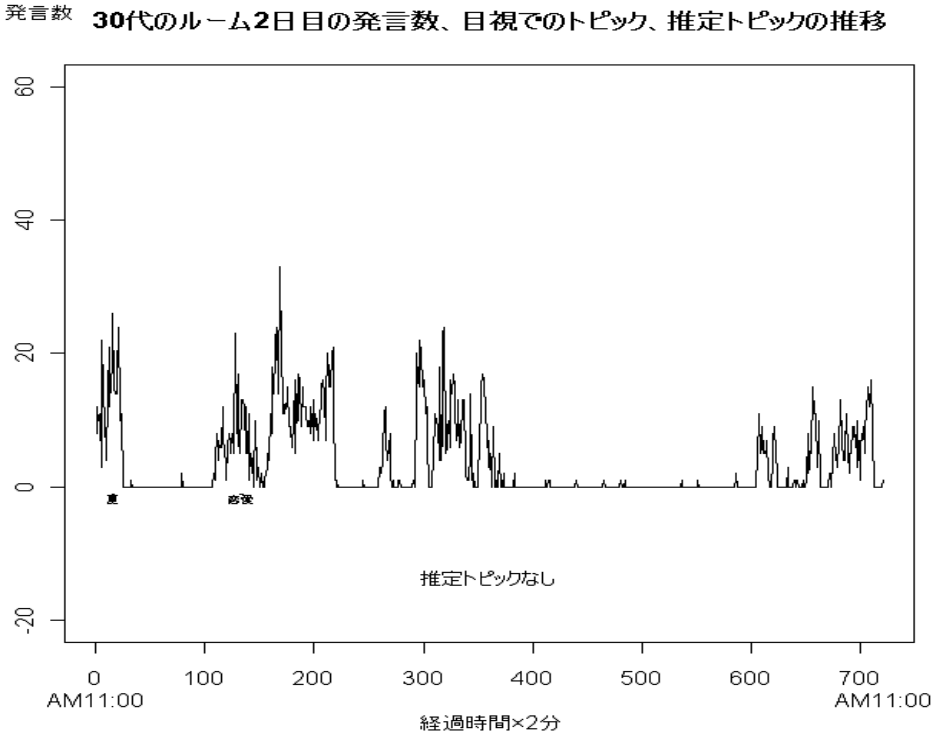


図 B.37: 30 代の 2 日目の発言数、目視でのトピック、推定トピックの推移

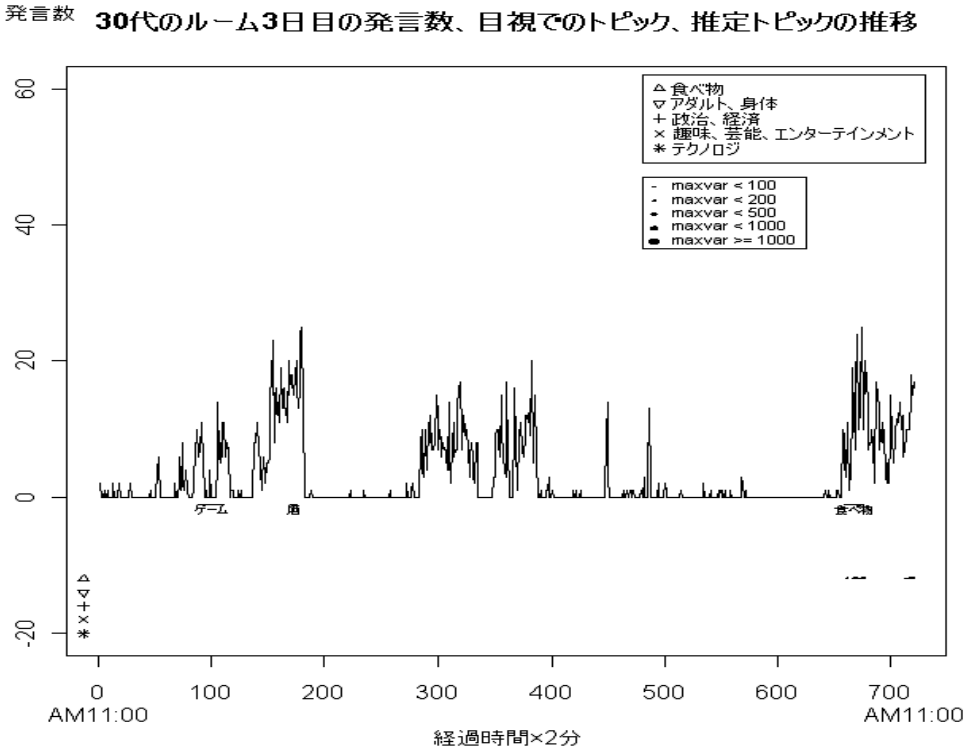


図 B.38: 30 代の 3 日目の発言数、目視でのトピック、推定トピックの推移

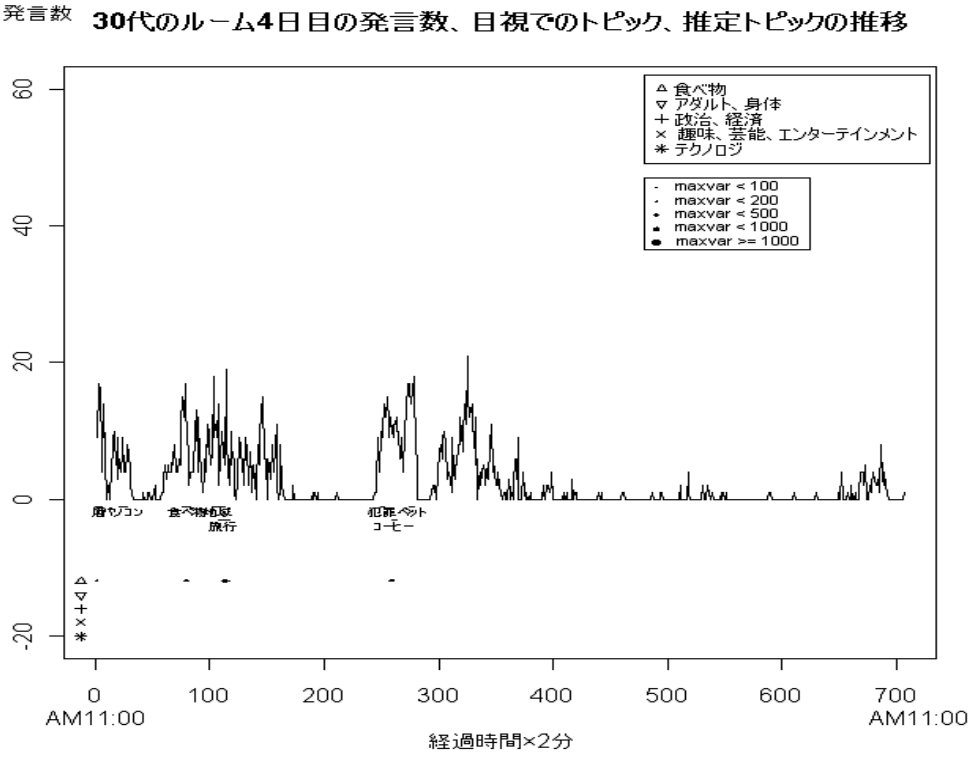


図 B.39: 30 代の 4 日目の発言数、目視でのトピック、推定トピックの推移

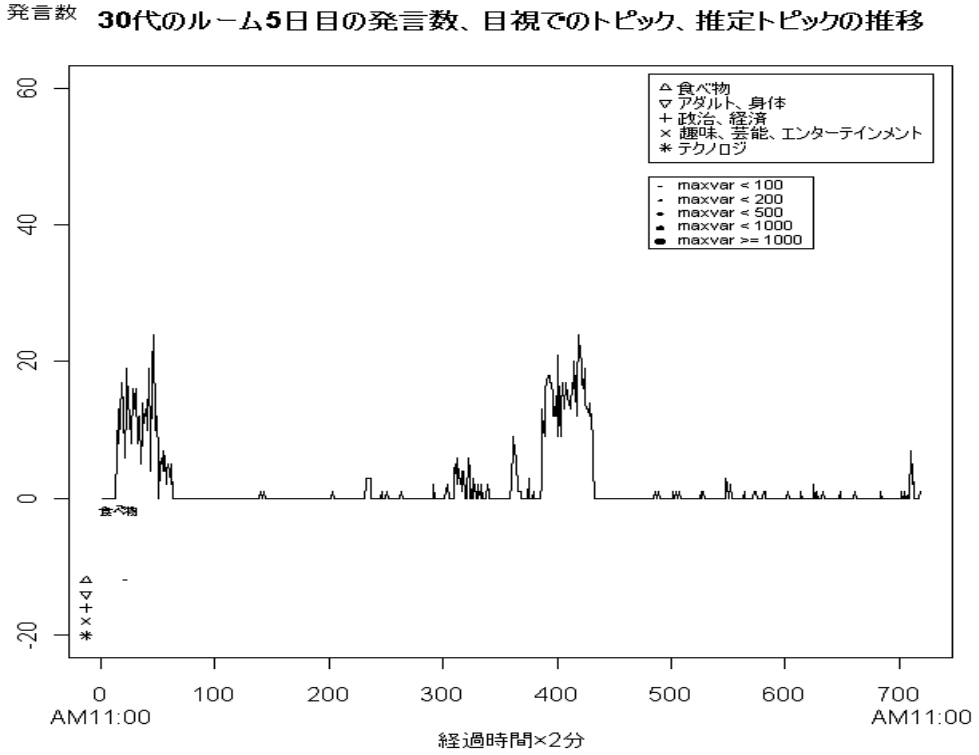


図 B.40: 30 代の 5 日目の発言数、目視でのトピック、推定トピックの推移

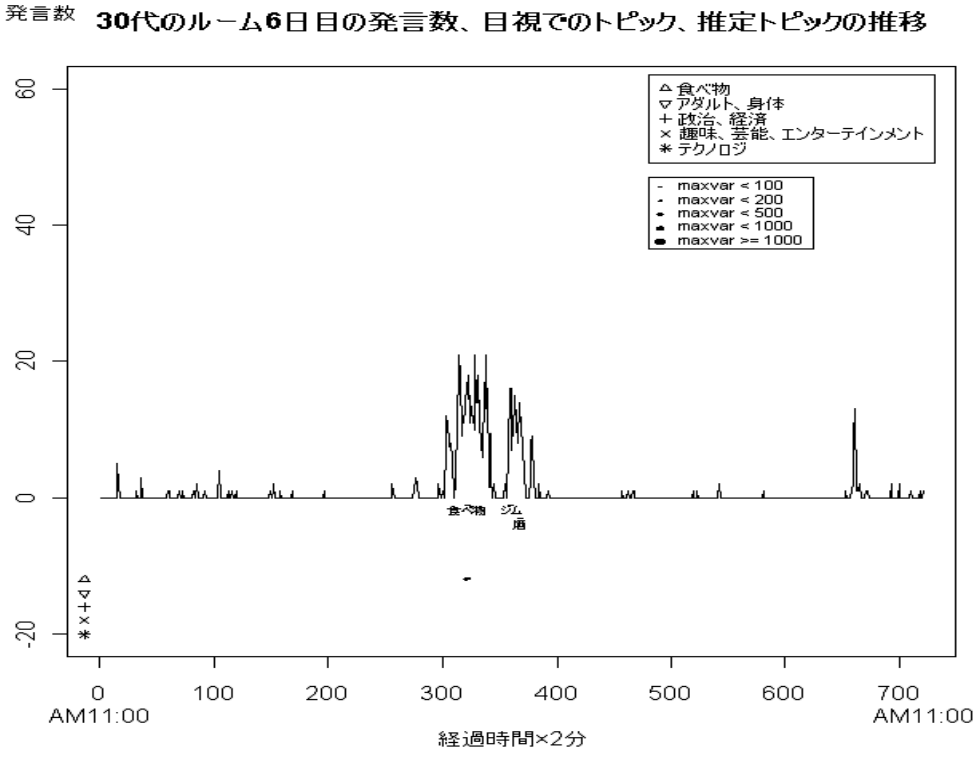


図 B.41: 30 代の 6 日目の発言数、目視でのトピック、推定トピックの推移

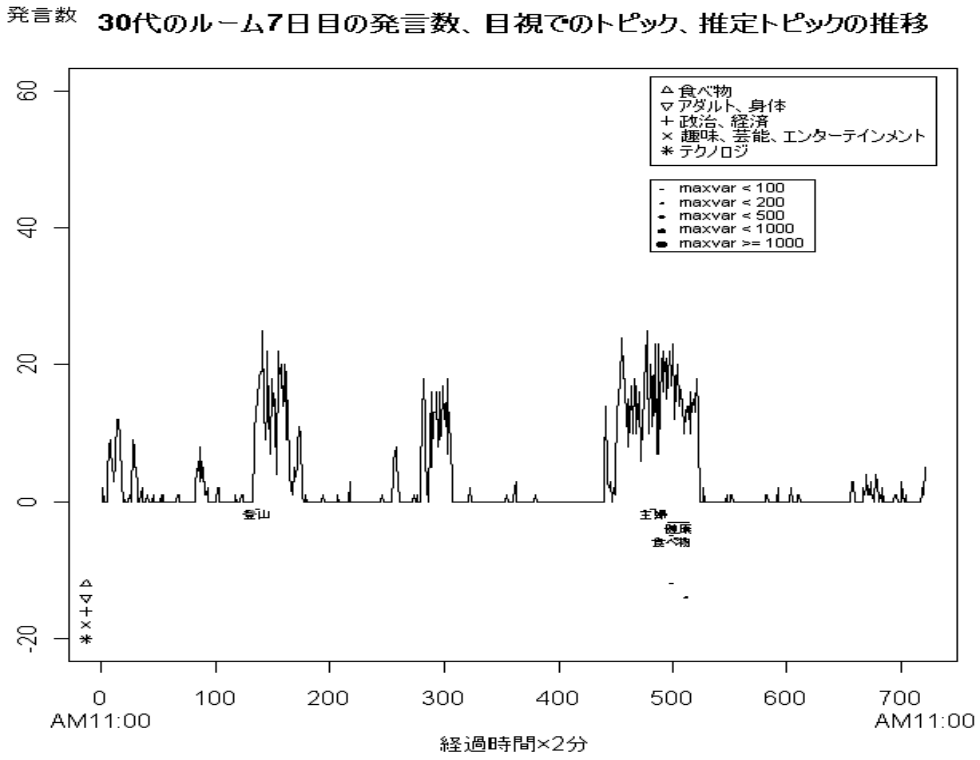


図 B.42: 30 代の 7 日目の発言数、目視でのトピック、推定トピックの推移

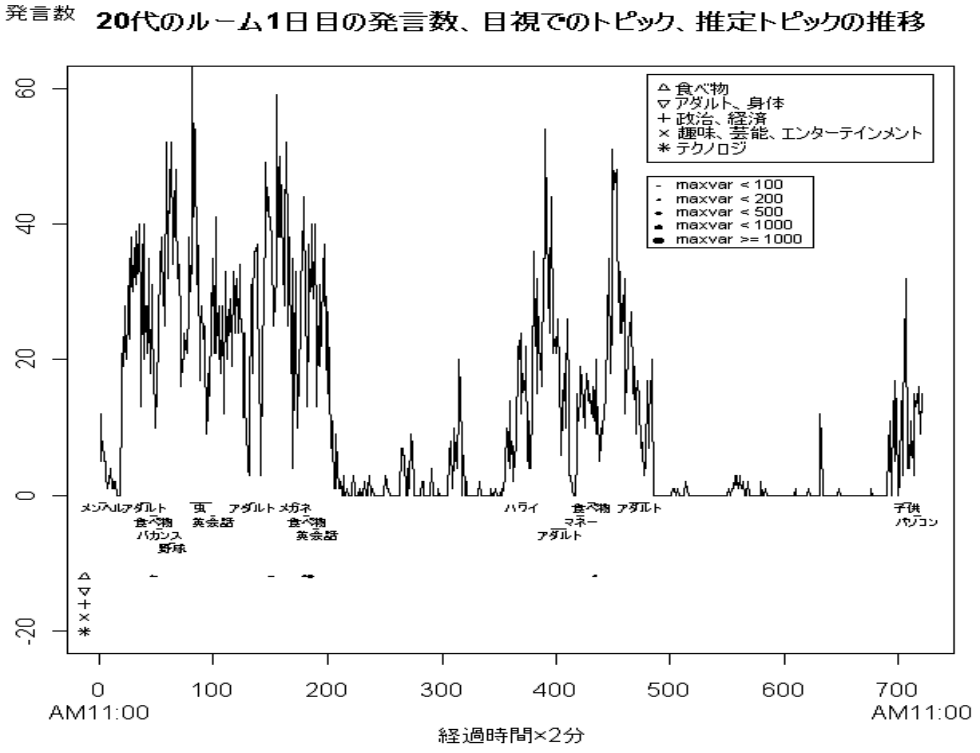


図 B.43: 20 代の 1 日目の発言数、目視でのトピック、推定トピックの推移

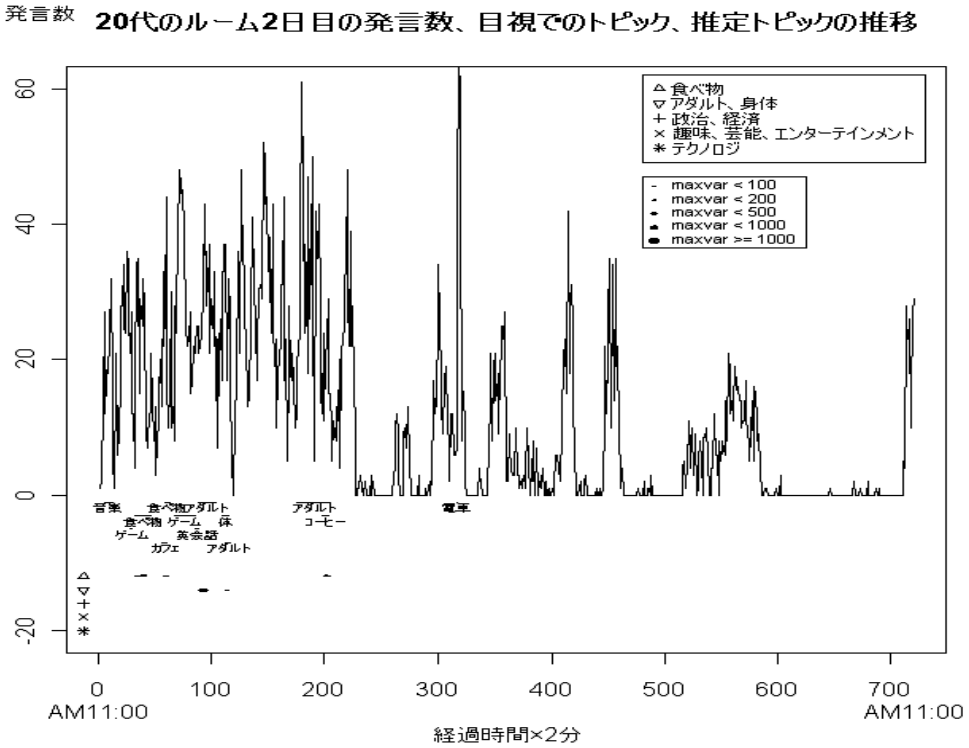


図 B.44: 20 代の 2 日目の発言数、目視でのトピック、推定トピックの推移

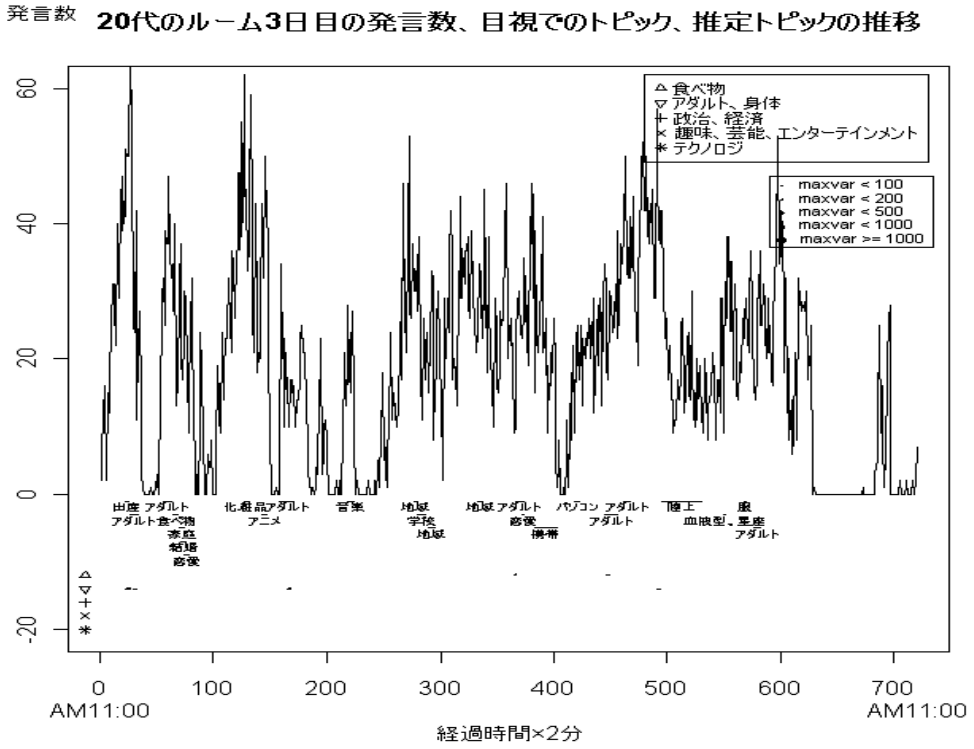


図 B.45: 20 代の 3 日目の発言数、目視でのトピック、推定トピックの推移

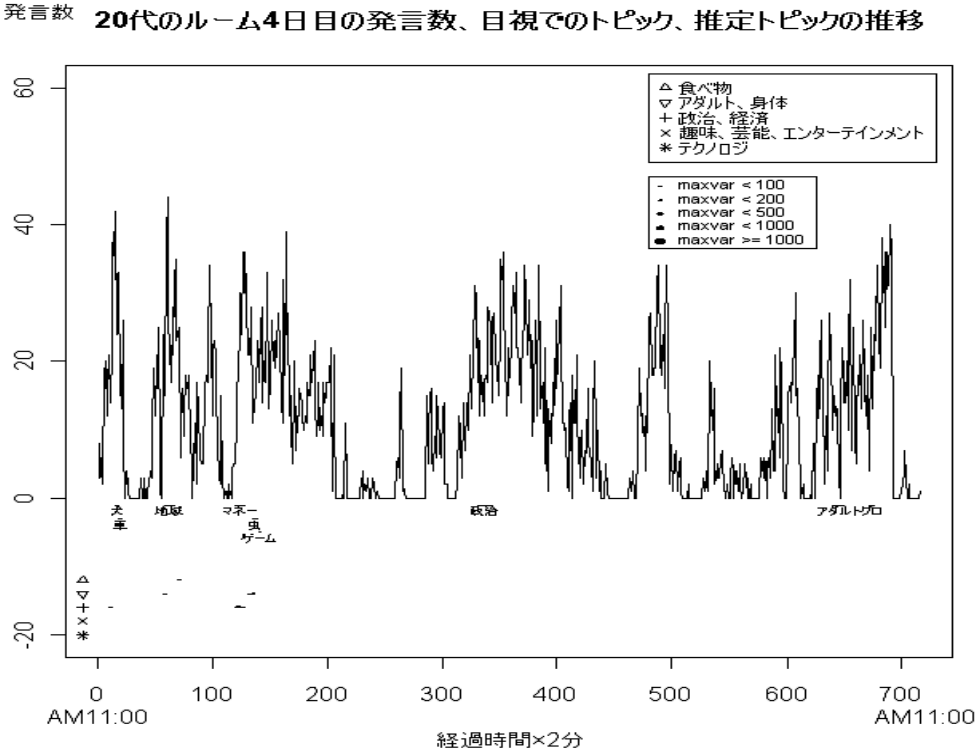


図 B.46: 20 代の 4 日目の発言数、目視でのトピック、推定トピックの推移

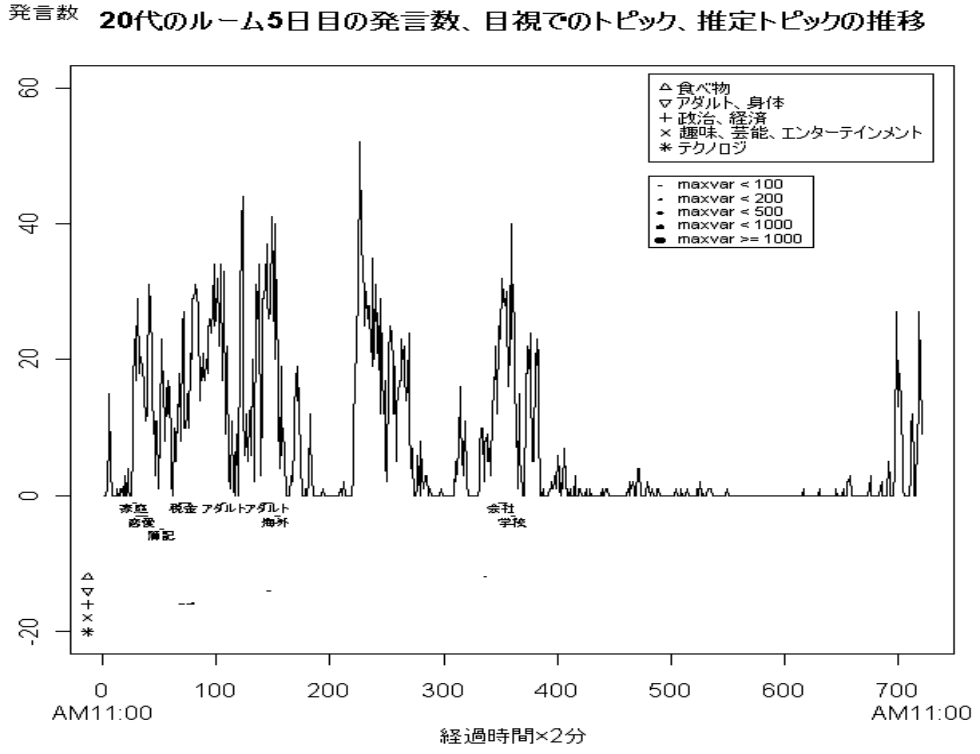


図 B.47: 20 代の 5 日目の発言数、目視でのトピック、推定トピックの推移

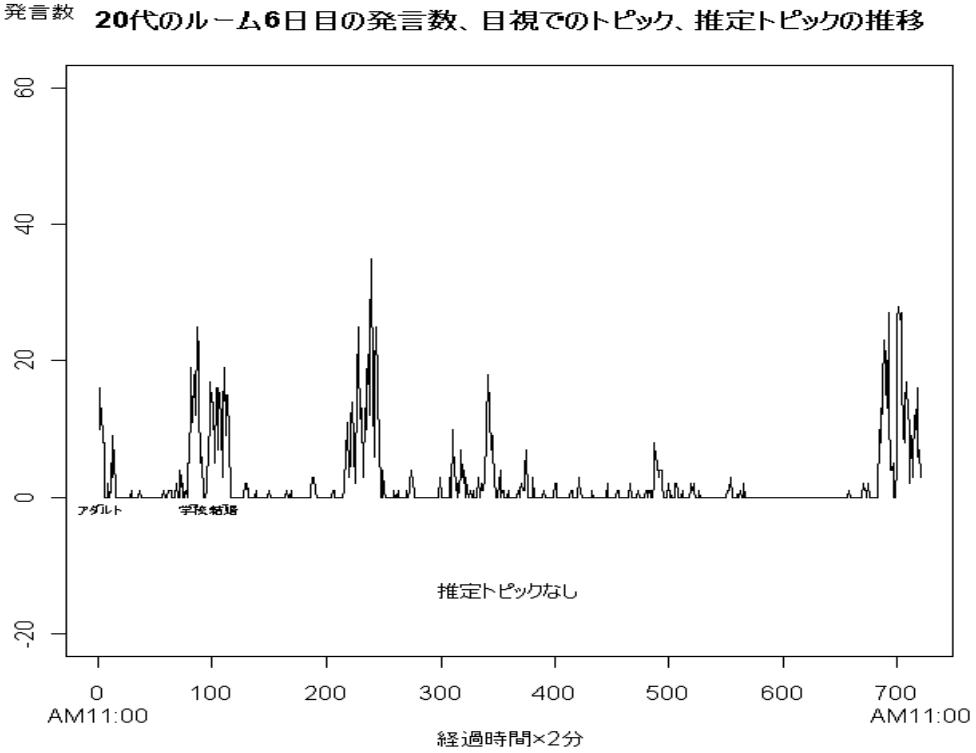


図 B.48: 20 代の 6 日目の発言数、目視でのトピック、推定トピックの推移

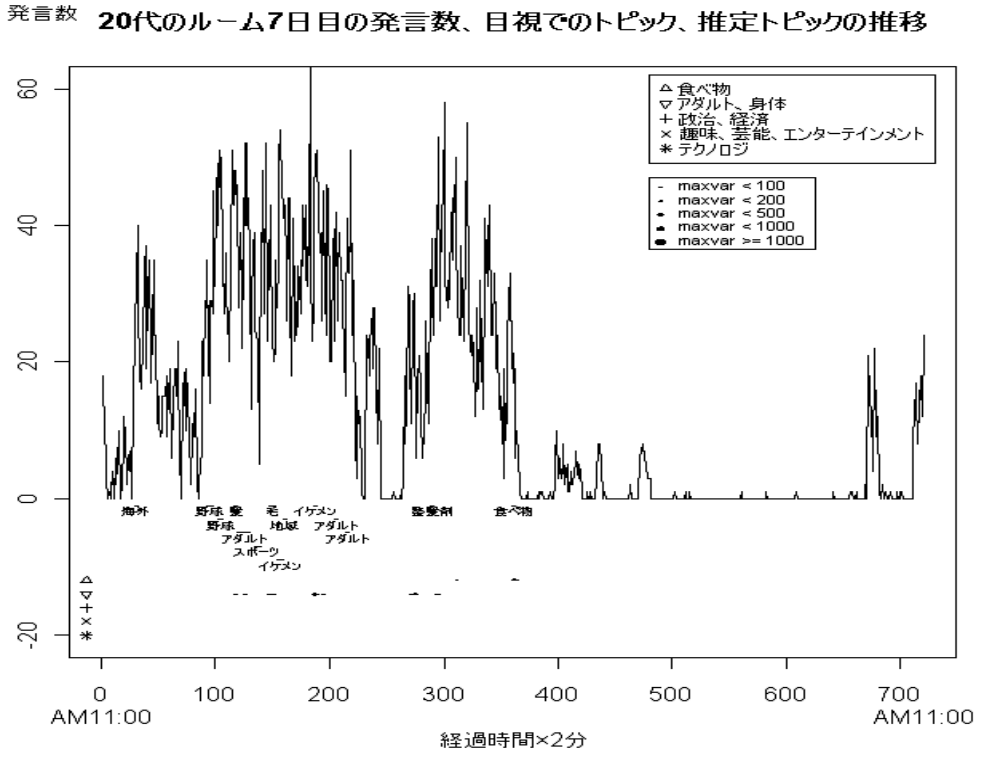


図 B.49: 20 代の 7 日目の発言数、目視でのトピック、推定トピックの推移