

YOUTUBE 配信者のための BERT-CRF を用いた 視聴者チャットのフィルタリング

岡田 雅大[†] 梶岡 慎輔[†] 山本 大介[†] 高橋 直久[†]

[†] 名古屋工業大学大学院工学研究科情報工学専攻 〒466-8555 名古屋市昭和区御器所町

E-mail: [†]masahiro@moss.elcom.nitech.ac.jp, ^{††}{kajioka,yamamoto,daisuke,naohisa}@nitech.ac.jp

あらまし 近年, YouTube をはじめとして様々な動画共有プラットフォームが普及している. そこではコンテンツを問わず, 様々な動画を誰でも見ることができる. その中でも配信者がリアルタイムで動画を配信することができる生放送では, 視聴者がチャットを送ることで配信者とのコミュニケーションが可能である. その場合, より視聴者のチャットを把握しやすくするために音声読み上げソフトを利用する配信者も存在する. しかし, 視聴者チャットが大量に流れてくる場合, 配信者は全てのチャットを把握することはできない. 特に音声読み上げソフトでは全てを読み上げることはできない. これらの問題を解決するために, 本研究では視聴者チャットの中から配信者が希望するチャットを BERT-CRF 等を用いて推定し, フィルタリングする手法を提案する. これにより配信者が希望するチャットを提示し, 視聴者と配信者のよりよいコミュニケーションを目指す. また, 提案手法の実験的評価によりラベリングの精度とリアルタイム性を評価する.

キーワード BERT, フィルタリング, 文タイプ, ラベリング, 機械学習

1 はじめに

現代において, YouTube やニコニコ生放送, Instagram をはじめとした動画共有プラットフォームが普及しており, 多くの人がコンテンツを問わず動画の生放送を行っている. その生放送では視聴者が配信者に対して文章を打ち込み, チャットとして配信者に届けることができる. 配信者はそのチャットを受け取り, 読み上げ, 反応することで視聴者と配信者のコミュニケーションが成り立つ. その中でも視聴者チャットをより把握するためにチャットを音声読み上げソフトに読み上げさせる配信者も存在する. だが, 視聴者チャットが大量に流れてくる場合, 配信者は全てのチャットを把握することはできない. 特に音声読み上げソフトでは全てを読み上げることはできない.

そこで本研究の目的は, 動画生配信者にとって有用なチャットのみを音声で提示することで視聴者と配信者のコミュニケーションを円滑にすることである. そのためには大量の視聴者チャットの中から配信者が希望するものを推定し, フィルタリングする必要がある. また, 動画生配信ではコミュニティによっては特有の言い回し等があり, そういった表現にも対応しなければいけない. 例えば, “～もろて” という表現があるが, これは“～してください”の意味である. そして音声読み上げソフトにおいては, リアルタイムに読み上げられる文量には制限があるため, 長文のようなチャットは, 視聴者が希望する箇所のみを抽出し, 短く読み上げる必要がある. この研究目的を目指し, 本研究では動画生配信のチャットに文タイプ推定手法を適用することで有用なチャットをフィルタリングすることを提案する.

文タイプ推定とは一つの文を機械的にタイプ分けする技術であり, 例えば“こんにちは”は“挨拶”と推定されるといった

文タイプの推定がある. 配信者が推定されたチャットの中から有用だと判断する文タイプを選択し, それに合わせてチャットを提示することで研究目的の達成を目指す. 提案システムのイメージは図 1 のようになる. 図 1 は配信者が数ある文タイプの内から“質問”タイプを選択した場合を示している. 例えば, “ここどこ?”というチャットは“質問”のラベルとして推定され, 読み上げ文として抽出された後に音声出力される.

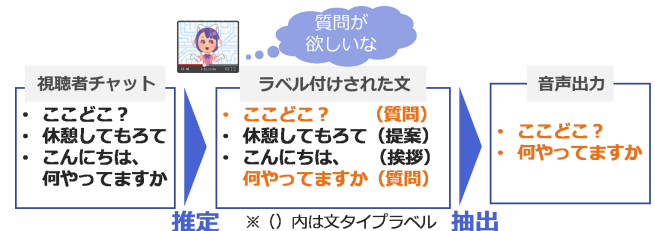


図 1 提案システムのイメージ (配信者が“質問”を選択した場合)

また, 従来の文タイプ推定の例を挙げると, Web お問い合わせ内容の質問タイプ分類 [1] では, 文タイプ推定を Web 上の実際の質疑応答サービスに適用し, お問い合わせ内容を質問タイプに分けた. 文タイプ推定を用いた感情分析精度の向上 [2] では文章をあらかじめタイプ分けした後に CNN によって感情分析を行うことで, 感情分析の精度向上を図った.

だが, 従来の文タイプ推定をそのまま本研究の目的に適用するには以下のような要件がある.

要件 1 従来の文タイプ推定は Web 上でのお問い合わせや一般文書等に適用されてきた. 例えばゲーム生配信では配信者が主人公であり, 配信者は視聴者に自分の言動への反応や感想を求めているのであって, 視聴者の自分語りなどを求めることは

少ない。配信者が必要だと考える文の種別分けが必要となる。

要件 2 動画生配信ではコミュニティによっては特有の言い回しがある。例えば、“～してもらて”という表現があるが、これは“～してください”という意味を示す。動画生配信の文タイプ推定を行うにはこれらの表現にも対応しなければいけない。

要件 3 動画生配信のリアルタイム性を考慮した文タイプ推定が必要である。例えば、生配信では長文で読み上げに時間のかかるようなチャットも存在する。長文に対しては読み上げ時間を短縮できるような工夫が必要である。

本研究では、上記の要件を満たす、配信者が希望するチャットを推定できる文タイプ推定手法を提案する。

2 関連研究

この章では、本研究の関連研究について述べる。

まず、研究 [3] では、CRF を用いた推定器を利用して音声対話システムにおける認識キーワードを推定した。一般のユーザが音声対話コンテンツ向けコンテンツを作成するためのシステム、MMDAgent では、エージェントが応答する文章である応答文と、応答文を引き出すためにユーザが発する語句である認識キーワードのセットを Web サイトを通じて登録することで、簡単に一問一答形式の音声対話コンテンツを作成することが可能である。しかし、機械との音声対話に慣れていない一般のユーザにとって、適切な認識キーワードを設定することは難しい。そこで、この研究では応答文と認識キーワードのペアを学習させたモデルを用いて認識キーワードを推定する手法を考えた。認識キーワード推定器の作成には Conditional Random Fields(CRF) が用いられたのだが、この研究の評価実験において TF-IDF 手法による認識キーワード推定と精度の比較を行い、CRF の方が高い精度を出している。

この研究 [3] は CRF を用いて固有表現抽出器を作成したが、本研究では同じく CRF を用いて文タイプ推定器を作成する。また、本研究では CRF だけでなく、Google の開発した BERT [16] を用いての文タイプ推定器作成も行う。

他の関連研究を挙げると、発話タイプ推定に関する研究 [4] では、ユーザ入力の発話タイプを推定した。訓練データには発話タイプラベルが付与された対話コーパスを用いている。発話タイプの推定には、まず事例類似度計算、特徴語抽出 (TF-IDF 法)、連鎖組織からスコアを算出して行われた。

この研究 [4] では人同士の雑談を対象とした一文単位の文タイプ推定を行っているが、本研究では CRF や BERT-CRF 手法を用いた視聴者チャットを対象とした単語系列単位の文タイプ推定を行う。

3 提案システムの概要

提案システムは以下のような特徴を持つ。

(1) 動画配信者のための文タイプラベル

動画配信者のための文タイプラベルを提案し、配信者は求めるチャットをラベルの中から選択する。今回はゲーム配信者を想定して、表 1 のようなラベルを考えたと、分野によって自由

な文タイプを設定することができる。表 1 のラベルの種類決定にはそれぞれゲーム配信者がチャットを欲しがらる場面の想定を行って決定した。これらにより、動画配信者にとって有用なチャットをフィルタリングする。

表 1 動画配信者のための文タイプラベル

文タイプラベル	例
FEEDBACK(フィードバック)	いいね
GREET(挨拶)	こんにちは
IMPRESSION(感想)	すごい
INFORMATION(情報提供)	～ですよ
PSN(人名)	～ちゃん
QUESTION(質問)	どうしたの
SUGGESTION(提案)	～しよう
O(上記のラベルに当てはまらないもの)	

(2) 動画生配信アーカイブにおいて流れたチャットの機械学習

動画生配信ならではの表現に対応するために、過去に行われた動画生配信で流れたチャットの機械学習を行う。今回はカバーストック株式会所属の YouTuber「兎田ぺこら」の過去配信で流れたチャットを取得し、機械学習させる。学習コーパスの一部を図 2 に示す。

```
4:17,ここから12時間やる気か.....!  
4:18,Oh she made a new stream  
4:21,i was confused  
4:22,peko Chan Suki desu  
4:22,lets gooo  
4:22,Is capcom paying her by the hour?  
4:22,12時間蹂躪されそう  
4:22,ここで腕にジャンプ攻撃だよ  
4:23,体壊さないようにね  
4:24,ohh were back  
4:25,good morning peko  
4:28,嘘ペコでしょ...、ちょっとは仮眠とってもらて  
4:28,まさかここから12時間かかることになるなんて  
4:29,クリアに一週間かかりそう  
4:33,12時間耐久すこすぎペコ  
4:34,ww  
4:34,草  
4:34,あかーん  
4:35,lol
```

図 2 選択した学習コーパスの一部

(3) 文タイプ推定の仕組みを単語系列単位で行えるように拡張したシステム

長文チャットは情報量が多く、複数の文タイプが存在する場合が多いと考えられる。一つのチャットに存在する複数の文タイプをそれぞれ分類し、必要な部分だけを抽出する事ができれば、長文の読み上げ時間を短縮することができる。例えば、“こんにちは、何やってますか?”という挨拶と質問の要素があるチャットにおいて、配信者が質問を求める場合、質問である“何やってますか?”の部分のみが抽出されると良い。

そして、一文複数の文タイプに対応するために、単語系列単位での文タイプ推定を提案する。具体的には、文章を形態素解析した後に、Conditional Random Fields(CRF) という機械学習モデルに基づいたラベリングによる文タイプ推定を行う。

提案システムの流れは図3のようになる。チャットデータを動画共有プラットフォームから取得し、データクレンジングにおいてチャットの事前処理を行う。処理したデータに形態素解析を行い、文タイプ推定器を用いて文タイプのラベリングを行う。ラベリングされたチャットから読み上げ文を抽出し、音声出力する。

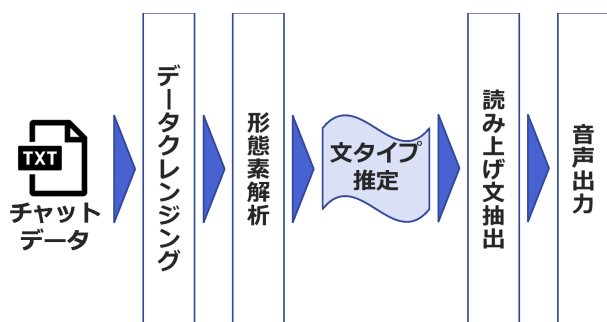


図3 提案手法の流れ

3.1 提案システムで用いる要素

3.1.1 形態素解析エンジン

形態素解析エンジンとは、自然言語処理の一種である形態素解析をプログラムによって自動的に行えるソフトウェアのことである。今回は Mecab と Juman++ の 2 種類の形態素解析エンジンを用いる。

a) Mecab

MeCab は、京都大学情報学研究科－日本電信電話株式会社コミュニケーション科学基礎研究所の共同研究ユニットプロジェクトを通じて開発されたオープンソース 形態素解析エンジンである。[13]

b) Juman++

Juman++は京都大学の黒橋研究室にて開発されている形態素解析エンジンである。言語モデルとして Recurrent Neural Network Language Model (RNNLM) を用いることにより、単語の並びの意味的な自然さを考慮した解析を行っている。それにより、JUMAN, MeCab に比べ大きく性能が向上している。[15]

3.1.2 機械学習モデル

本研究では文タイプ推定に機械学習を行うが、今回は比較のため Conditional Random Fields(CRF) と BERT-CRF の 2 種類の機械学習モデルを用いる。

a) Conditional Random Fields(CRF)

CRF は、確率的グラフィカルモデルの一つであり、識別モデルである。特に、系列ラベリングを解くために使用される。系列ラベリングとは、入力に単語列を受け取り、出力として個々のデータにラベルを付与することが目的となる問題である。また、識別モデルとは、各クラスごとに各要素が分類される確率を計算し、問題を解く手法である。そして、CRF では構造学習が取り入れられており、例えば、単語列が渡された場合、個々の単語一つ一つに分割して処理をせず、単語列として纏めて処理をするため、単語の前後関係を判断できる。

b) BERT-CRF

BERT-CRF は、事前学習済みの BERT モデルの最下層に CRF を追加し、正解ラベルを出力するように Fine-tuning を行ったものである。また、この BERT (Bidirectional Encoder Representation from Transformer) とは、2018 年 10 月に Google から発表された自然言語処理モデルである。その特徴として、Transformer という構造が組み込まれており、文章を双方向から学習することによって、文脈を読むことが可能である。また、今回利用する BERT の言語モデルが京都大学の黒橋研究室の日本語 Pretrained モデル [19] であり、形態素解析に Juman++ を推奨しているため、CRF とは形態素解析器が異なる。

4 提案システムの実現法

本章では提案システムの実現法について記述する。

4.1 チャットの取得

YouTube からチャットの取得を行う。Youtube Live Streaming API [18] では現在配信中のチャットは取得できるが、配信アーカイブのチャットの取得はできない。今回は実際に行われている生配信に対するリアルタイムシステムの実装を行う訳ではないため、チャットの取得にはクローリングを行う。クローリングとはプログラムがインターネット上のリンクを辿って Web サイトを巡回し、Web ページ上の情報を取得することである。今回は特定のゲーム配信アーカイブの URL を指定し、チャットの取得を行う。

4.2 文タイプ推定器の作成

文タイプ推定器を作成するための流れを図4に示す。また、それぞれの内容について 4.2.1-4.2.4 節で説明する。

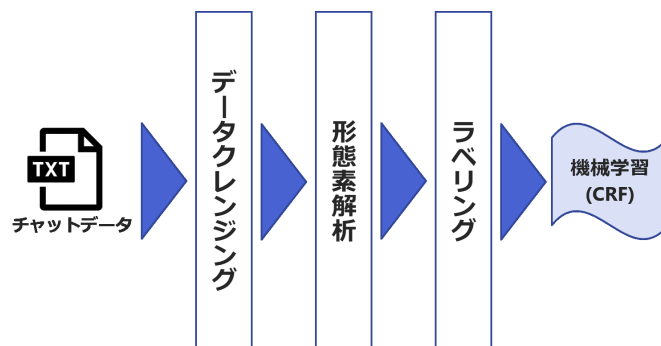


図4 提案システム実現の流れ

4.2.1 データクレンジング

プログラムによってチャットデータの整形を行う。チャット以外のデータ部分の削除、日本語以外のチャットの削除、機械学習に支障をきたす文字列 (" ", ". ", "# "等) の削除は??節で示した事と同様に行う。ここでは機械学習を行うので、上記で示した事に加えてデータの重複排除を行う。つまり、同じ内容のチャットの削除を行う。

データクレンジング前のチャットデータを図5にデータクレ

ンジング後のチャットデータを図 6 に示す。

```
-0:03,配信上限は12時間ガ
-0:03,にーん...
-0:02,こんなに長い案件配信になると誰が予想したのか・・・
-0:02,がんばって
-0:02,this reminds me the minecraft RTA
-0:01,PEKO PEKO PEKO PEKO
-0:01,応援してるよー!!!
-0:01,マジかあ!べこちゃん!頑張れ
-0:01,Been here since the start Peko
-0:00,ちょっと休んでもいいのよ
0:02,There is issues with the serverso you have to end at 12 usually
0:02,にーん...
0:02,peko
0:02,意地を見届けたいべこ
0:04,べこらの戦いはこれからだ!!!次回作にご期待ください
0:05,What Happened
0:06,久しいな12時間以上耐久
```

図 5 データクレンジング前

```
配信上限は12時間ガ
にーん...
こんなに長い案件配信になると誰が予想したのか・・・
応援してるよー!!!
マジかあ!べこちゃん!頑張れ
ちょっと休んでもいいのよ
意地を見届けたいべこ
べこらの戦いはこれからだ!!!次回作にご期待ください
久しいな12時間以上耐久
毎日の配信を楽しみにしてるのにこれで体調崩されたら...
にーん...
ケツイを力に変えるんだ!
本当に今月中にクリアできない可能性出てきた
こんべこ2!
行きそう
スラの魔界村で耐久とはさすが長だなw
始まる前はまさか耐久とは
```

図 6 データクレンジング後

4.2.2 形態素解析

クレンジングされたデータに形態素解析を行う。今回は Mecab と Juman++を用いる。これら 2 つを用いる理由は、今回利用する CRF モデルと BERT の言語モデルがそれぞれ Mecab と Juman++を推奨しているためである。Mecab と Juman++で形態素解析を行った後のチャットデータをそれぞれ図 7、図 8 に示す。

```
こんなに 副詞 助詞類接続 * 形容詞・アウオ段 * 基本形 長
長い 形容詞 目立 * 形容詞・アウオ段 * 基本形 長
案件 名詞 一般 * 案件 * アンケン
配信 名詞 格助詞 一般 * 配信 * ハニナ
に 助詞 自立 * 五段・ラ行 * 二なる
なる 助詞 接続助詞 * 基本形 * タレ
と 助詞 代名詞 一般 * 誰か * タガ
誰か 名詞 格助詞 一般 * 誰か * ガ
予想 名詞 自立 * サ変・スル * 連用形
した 助詞 非自立 一般 * 特殊・タ * 基本形
のか 助詞 副助詞/並立助詞/終助詞 * の * ナン
か 助詞 一般 * * *
. 助詞 一般 * * *
. 助詞 一般 * * *
```

図 7 Mecab による形態素解析

```
こんなに 副詞 助詞類接続 * 指示詞 副詞形態指示詞 * 代
長い 形容詞 目立 * 形容詞・アウオ段 * 基本形 代
案件 名詞 一般 * 案件 * アンケン
配信 名詞 格助詞 一般 * 配信 * ハニナ
に 助詞 自立 * 五段・ラ行 * 二なる
なる 助詞 接続助詞 * 基本形 * タレ
と 助詞 代名詞 一般 * 誰か * タガ
誰か 名詞 格助詞 一般 * 誰か * ガ
予想 名詞 自立 * サ変・スル * 連用形
した 助詞 非自立 一般 * 特殊・タ * 基本形
のか 助詞 副助詞/並立助詞/終助詞 * の * ナン
か 助詞 一般 * * *
. 助詞 一般 * * *
. 助詞 一般 * * *
```

図 8 Juman++による形態素解析

4.2.3 ラベリング

形態素解析したテキストデータに手入力では表 1 のラベルを当てはめる。ラベルを付与する方法は下図 9 に示すように、形態素解析されたテキストデータにおいて、形態素解析情報の tab を挟んで右端にラベルを付け加える。Mecab と Juman++では

形態素解析結果は異なるが、同一の箇所に同一のラベルを付与する。配信者のためのラベルの種類決定にはそれぞれ配信者がチャットを欲しがらる場合の想定を行って決定した。FEEDBACK は視聴者の反応がほしい場合、GREET は新規視聴者が入ってきたときに反応したい場合、IMPRESSION は視聴者の感想を聞きたい場合、INFORMATION は配信者が行っている作業に関する情報やアドバイスを仕入れたい場合、PSN は他の配信者の話題などがほしい場合、QUESTION は質問が欲しい場合、SUGGESTION は視聴者から指示がほしい場合を想定した。他にも想定される場面とラベルは考えられるだろうが、今回はこれらのラベルの推定器を作成する。

より具体的にラベルの付与方法を説明すると、まず GREET、PSN については一つのあいさつと名前を単位としてラベリングを行う。例えば、”こんにちは”，”さようなら”等に GREET ラベル，”岡田君”，”ヒカキンさん”等に PSN ラベルが付与される。一方で FEEDBACK、IMPRESSION、INFORMATION、QUESTION、SUGESTION は意味の通じる単位を考慮してラベリングを行う。例えば、”そうだよ”，”ちがうよ”等に FEEDBACK，”安心した”，”やばすぎて草”等に IMPRESSION ラベル，”ボスが出る”，”川がある”等に INFORMATION ラベル，”多くない? ”，”寝ましたか?”等に QUESTION ラベル，”止まらない方がいい”，”寝てくれ”等に SUGGESTION ラベルが付与される。

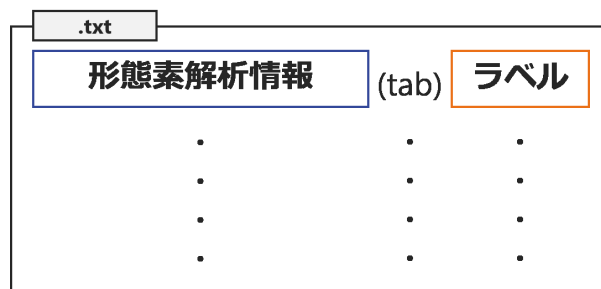


図 9 ラベル付与方法

4.2.4 機械学習

機械学習は 3.1.2 節で説明した学習モデルの CRF と BERT-CRF を用いる。それぞれの学習モデルが推奨する形態素解析と組み合わせて、Mecab+CRF、Juman+BERT-CRF の 2 通りで行う。ラベリングされたチャットデータを機械学習することで、文タイプ推定器の作成を完了する。

4.3 読み上げ文の抽出

文タイプ推定がされたチャットの内、選択されたラベル部分を抽出し、読み上げ文とする。ラベルが付与された部分のみを抽出するだけでは、ラベルの推定ミスによる単語の抜けや、主語や重要な修飾部分が含まれず、チャットの内容が十分に伝わらない等の問題が予測されるため、以下の規則に基づいて読み上げ文の抽出を行う。番号の小さいものほど優先度が高い。また、音声読み上げにおいて文の間に区切りを作るため、文の終わりには”。”を付け加える。最後に、抽出した読み上げ文を txt ファイルに出力する。

(1) 文内にユーザによって選択されたラベルが存在しない場合、抽出しない

(2) ユーザによって選択された文タイプラベルが存在した場合、これに対応する単語を抽出する

(3) ”2”で抽出された単語の後列が O ラベルの場合、後列の特定の記号（!, ?, , , . など）またはユーザに選択されていないラベルの一つ前までを”2”と同様のラベルとして抽出する

(4) ”2”で抽出された単語の前列が O ラベルの場合、前列の特定の記号（!, ?, , , . など）またはユーザに選択されていないラベルの一つ後までを”2”と同様のラベルとして抽出する

上記の規則に沿って読み上げ文を抽出した例を下表 2 に示す。そのまま推定結果を抽出してしまうと”壊さないようにね”という文になり、何を壊さないように言っているのか伝わらない。そこで上記の規則に沿った抽出では”体壊さないようにね”と抽出され、意味が正確に伝わる。

表 2 読み上げ文の抽出例 2

項目	単語・ラベル (SUG は SUGGESTION ラベルを示す)					
チャット	体	壊さ	ない	よう	に	ね
推定結果	O	SUG	SUG	SUG	SUG	SUG
抽出結果	SUG	SUG	SUG	SUG	SUG	SUG

4.4 出力

抽出した読み上げ文を Open JTalk によって読み上げる。具体的には、4.3 節で出力した txt ファイルを Open JTalk に通して、wav ファイルを生成する。生成された wav ファイルを音楽プレーヤー等で再生することで、読み上げ文が音声出力される。

5 提案手法の評価

提案手法の評価を行うためにに行った 2 つの評価実験について記述する。

5.1 評価実験 1 (精度検証)

5.1.1 実験の目的

Mecab × CRF と Juman++ × BERT-CRF それぞれの文タイプ推定器の性能を検証すること。ラベリングの推定精度評価を行う。あらかじめ手作業でラベリングした正解データと推定結果を比較し、適合率、再現率、F 値等の値を求める。

5.1.2 実験方法

Mecab+CRF ではデータを 10 分割し、1 割をテストデータ、8 割を学習データとする。Juman+BERT-CRF では検証データを用いて学習を繰り返す必要があるため、データを 10 分割し、1 割をテストデータ、1 割を検証データ、8 割を学習データとする。それぞれの場合でラベルごとに precision(適合率)、recall(再現率)、f1-score(F 値) を計算し、クロスバリデーションを行う。それによって得られた数値から重み付き平均を計算し、結果をグラフ化する。

5.1.3 結果と考察

Mecab+CRF の結果を図 10、Juman+BERT-CRF の結果を図 11 に示す。

Mecab+CRF の場合では”GREET”と”PSN”は再現率と F 値が少し小さい値を示したが、全体を通して高水準なスコアを得られた。文タイプ推定として良好な性能を示した。

Juman+BERT-CRF の場合は、Mecab+CRF に比べて上回ったのは’PSN’の再現率と F 値のみであり、それ以外の値は全て下回った。そのため、言語モデルや形態素解析器を Mecab に変更して実験を行いもしたが、精度の向上は見られなかった。原因は定かではないが、考えられる原因としてデータ不足による過学習等が考えられる。

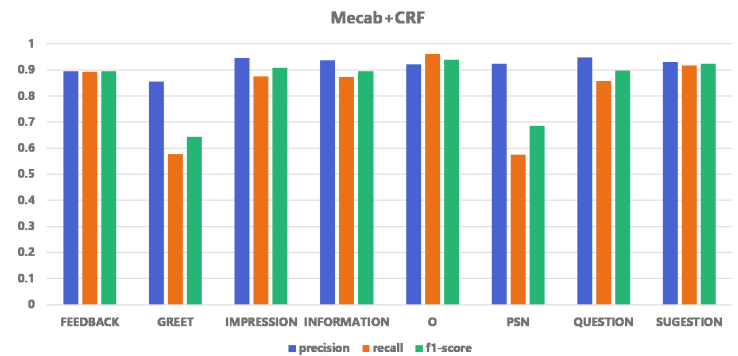


図 10 Mecab+CRF の推定結果

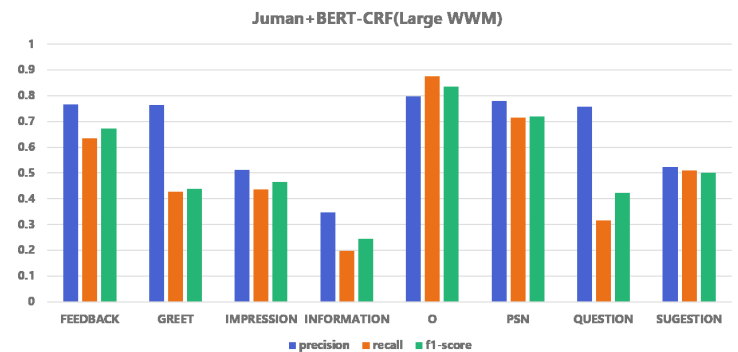


図 11 Juman+BERT-CRF の推定結果

5.2 評価実験 2 (リアルタイム性の検証)

5.2.1 実験の目的

動画配信者と視聴者のコミュニケーションをリアルタイムで間に合うように文タイプ推定を行うことが可能であることを検証すること。

5.2.2 実験方法

チャット取得プログラム、データクレンジングプログラム、形態素解析プログラム (Mecab と Juman++)、文タイプ推定プログラム (CRF と BERT-CRF によって学習済み)、それぞれの処理にかかる時間を計測する。計測は Java には current-TimeMillis 関数、Python には time 関数を用いて行う。

5.2.3 結果と考察

実験結果を下表 3 に示す．時間の単位は秒であり，小数点以下第 5 位を四捨五入している．また，処理時間は一文当たりの処理にかかった時間を計測している．

Mecab+CRF では一つのチャットを処理するのにかかる時間は合計 0.0343 秒であり，1 秒に 29 個のチャットが流れてきても問題ないという計算になる．一方で Juman+BERT-CRF では BERT-CRF の処理時間が大きく，一つのチャットの処理にかかる時間は合計 0.1768 秒となった．これは 1 秒に 6 個以上のチャットが流れてくると処理が間に合わない計算になる．

よって，処理時間については，Mecab+CRF 手法が優れていると考えられる．

表 3 処理時間

処理内容	時間 (秒)
チャット取得	0.0154
データクレンジング	0.0002
Mecab	0.0185
CRF	0.0001
読み上げ文抽出	0.0001
計 (Mecab+CRF)	0.0343
Juman++	0.0016
BERT-CRF	0.1596
読み上げ文抽出	0.0001
計 (Juman++BERT-CRF)	0.1768

6 評価実験 3(読み上げの検証)

6.1 実験の目的

Open JTalk を用いてチャットの読み上げにかかる時間を計測し，提案手法である単語系列単位の文タイプ推定が全文読み上げ手法や文単位の文タイプ推定手法と比較してリアルタイム性においてどのくらい優れているのかを検証すること．また，チャットの内容が音声で十分に伝わるかを検証すること．

6.2 実験方法

以下の 5 つの手法による Open JTalk を用いた読み上げ時間を計測し，比較を行う．読み上げに用いるチャットデータはゲーム配信 (本編 803 秒) で流れたチャット約 1300 文にデータクレンジングシステムを実行した後の約 650 文を用いる．ここでは QUESTION ラベルを対象を絞って実験を行う．また，それぞれの手法で読み上げる文量についても調べる．そして，読み上げ内容についても，チャットの内容が正確に伝わるか，内容が選択したラベルのものであるかを確認する．

- チャットの全文を読み上げる (以下”全文読み上げ”)
- 提案手法である Mecab+CRF の文タイプ推定器によってラベリングされたチャットの内，QUESTION ラベルが一つでも付与されたチャットの一文全てを読み上げる (以下”Mecab+CRF 文単位”)
- 提案手法の実現法通りに Mecab+CRF 手法で文タイプ推定を行い，QUESTION ラベルに対して読み上げ文を抽出し

て読み上げる (以下”Mecab+CRF プロト”)

- 提案手法である Juman+BERT-CRF の文タイプ推定器によってラベリングされたチャットの内，QUESTION ラベルが一つでも付与されたチャットの一文全てを読み上げる (以下”Juman+BERT-CRF 文単位”)
- 提案手法の実現法通りに Juman+BERT-CRF 手法で文タイプ推定を行い，QUESTION ラベルに対して読み上げ文を抽出して読み上げる (以下”Juman+BERT-CRF プロト”)

6.3 結果と考察

5 つの手法による文字数と読み上げ時間についてそれぞれ表 12，表 13 に示す．読み上げ時間の単位は秒であり，全文読み上げは 6724 文字で 1114 秒，Mecab+CRF 文単位は 372 文字で 58 秒，Mecab+CRF プロトは 327 文字で 53 秒，Juman+BERT-CRF 文単位は 351 文字で 60 秒，Juman+BERT-CRF プロトは 224 文字で 41 秒読み上げにかかったことが分かる．全文読み上げに関しては，本編時間 803 秒を大幅に超えてしまった．やはり全てのチャットを読み上げることは難しいことが分かる．次に，Mecab+CRF について見てみると，文単位に比べて Mecab+CRF プロトは文字数 45，読み上げ時間 5 秒が削減されたことが分かる．5 秒あれば短い文だと 3~4 文読み上げることが可能であるので有意的な削減とだと考えられる．また，質問と推定された 30 文の内，推定ミスにより意味が十分に伝わらないものは一文だけであった．最後に Juman+BERT-CRF について見ると，CRF と比べて読み上げ時間と文字数だけに関すれば優れていると見える．しかし，Juman+BERT-CRF に関しては推定ミスが多く，読み上げ内容が質問ですらないものが多かった．特に，BERT-CRF プロトの読み上げ内容は”?”や”か”など，まったく意味の通じないものが多かった．

これらから，BERT-CRF に関しては文タイプ推定として上手く機能しなかった．一方で，CRF に関しては推定ミスも少なく，ほとんどが正確に意味の通じるものが多く，また読み上げ時間も有意的に削減出来ており，リアルタイム性に優れた文タイプ推定ができると考えられる．

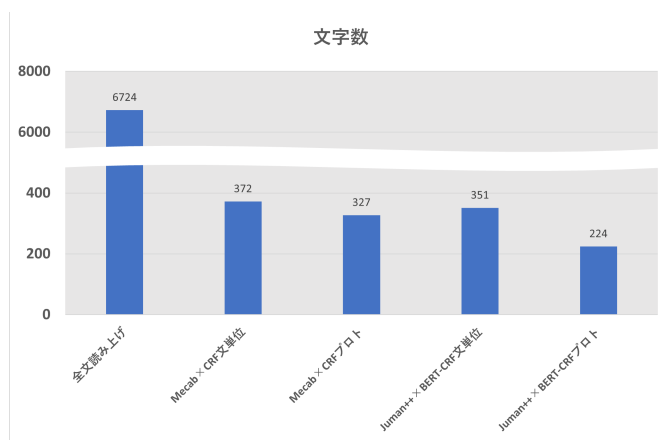


図 12 文字数

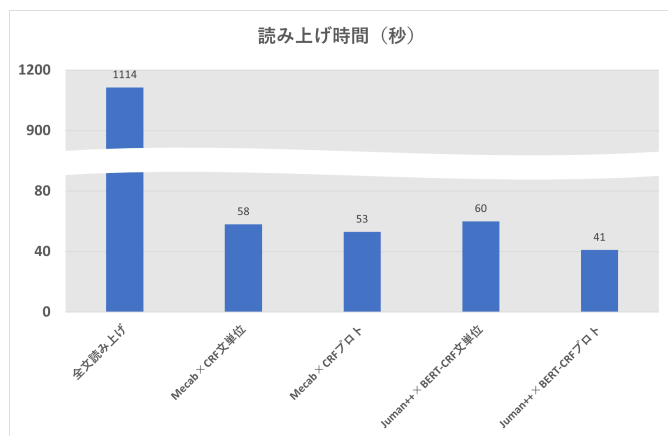


図 13 読み上げ時間

7 おわりに

動画配信者と視聴者のコミュニケーションを円滑にするために動画配信者が希望する視聴者チャットをフィルタリングするシステムについて記述した。システムの特徴として、配信者のための文タイプラベル、動画生配信アーカイブにおいて流れたチャットの機械学習と文タイプ推定の仕組みを単語系列単位で行えるように拡張したシステムを提案した。また、提案システムの評価実験を行ったが、今回のゲーム生配信動画に提案した文タイプラベルの推定を適用させることを考えると、Juman+BERT-CRF よりも Mecab+CRF の方がラベリング精度、処理時間、読み上げ内容に関して上回っているという結論となった。しかし、動画コミュニティが変わればそれに合わせた文タイプ種別分けが必要となってくるので、一概に動画生配信に文タイプ推定を適用させるには Mecab+CRF の手法が Juman+BERT-CRF よりも優秀であるとは言えない。また、チャットのデータ数が増えれば BERT-CRF の過学習問題等も解消され、より精度に優れた文タイプ推定ができる可能性もある。この点は今後の課題とする。今後の課題を他にも述べると、リアルタイムシステムの実装、実際の配信での試験運用等も挙げられる。

文 献

- [1] M. Takeuchi, T. Tokunaga and Y. Matsumoto, “Chunking-based Question Type Identification for Multi-Sentence Queries”, in Proceedings of the SIGIR 2007 Workshop on Focused Retrieval, pp.41–48, 2007.
- [2] T. Chen, R. Xu, Y. He, X. Wang, “Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN”, Expert Systems with Applications, Vol. 72, pp.221-230, 2017.
- [3] 前田 一樹, “固有表現抽出を用いた音声対話コンテンツ向け認識キーワードの推定手法”, 名古屋工業大学修士論文, 2019.
- [4] 瀬戸 一馬, 岸 義樹, “隣接関係による発話タイプ推定を用いた対話システムの構築”, 第 77 回全国大会講演論文集, pp.131-132, 2015.
- [5] Zhenjin Dai, Xutao Wang, Pin Ni, Yuming Li, Gangmin Li, Xuming Bai, “Named Entity Recognition Using BERT BiLSTM CRF for Chinese Electronic Health Records”,

- 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019.
- [6] Alexandros Komninos, Suresh Manandhar, “Dependency Based Embeddings for Sentence Classification Tasks”, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.1490–1500, 2016.
- [7] 高村 大也, 松本 裕治, “SVM を用いた文書分類と構造的帰納学習法”, 情報処理学会論文誌データベース (TOD), pp.1-10, 2003.
- [8] 山田 寛康, 工藤 拓, 松本 裕治, “Support Vector Machine を用いた日本語固有表現抽出”, 情報処理学会論文誌, pp.44-53, 2002.
- [9] 江村 優花, 関 洋平, “テキストに現れる感情, コミュニケーション, 動作タイプに基づく顔文字の推薦”, IPSJ SIG Technical Report, 2012.
- [10] 青柳 達也, 山本 幹雄, 板橋 秀一, “韻律情報を用いた発話文タイプ付与支援システム”, 全国大会講演論文集, pp.229-230, 1997.
- [11] D. Yamamoto, S. Tsutsumi and T. Uchiya, “Web-based user generating environment for voice dialogue content and its evaluation experiments”, IEICE technical report, pp. 111-116, 2015.
- [12] D. YAMAMOTO, T. MASUDA, S. OHIRA, K. NAGAO, “Video Scene Annotation Based on Web Social Activities”, IEEE Multimedia, Vol.15, No.3, pp.22-32, 2008.
- [13] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, <https://taku910.github.io/mecab/> 2013-02-18.
- [14] mecab-ipadic-NEologd : Neologism dictionary for MeCab, <https://github.com/neologd/mecab-ipadic-neologd/blob/master/README.ja.md> 2020-08-25.
- [15] 日本語形態素解析システム Juman++, <https://nlp.ist.i.kyoto-u.ac.jp/?JUMAN%2B%2B> 2021-11-19.
- [16] BERT とは, <https://ledge.ai/bert/> 2020-06-12.
- [17] 五位野 琢也, 濱上 知樹, “BERT を用いた医療文書からの固有表現抽出”, 2021.
- [18] YouTube Live Streaming API, <https://developers.google.com/youtube/v3/live/getting-started>(2021.1.12 参照).
- [19] BERT 日本語 Pretrained モデル, https://nlp.ist.i.kyoto-u.ac.jp/?ku_bert_japanese 2020-11-21.
- [20] BERT-CRF, <https://github.com/Louis-udm/NER-BERT-CRF> 2021-03-09.
- [21] 【チュートリアル】機械学習を使って 30 分で固有表現抽出器を作る, <https://qiita.com/Hironan/items/326b66711eb4196aa9d4> 2017-05-03.
- [22] Open JTalk, <http://open-jtalk.sp.nitech.ac.jp/> (2022.1.14 参照).
- [23] HuggingFace, https://huggingface.co/docs/transformers/model_doc/bert(2022.1.25 参照).