

話題候補を実時間提示する 個人生放送支援システムの実装と評価

桐戸 創也[†] 山本 景子[†] 倉本 到[†]
辻野 嘉宏[†] 水口 充[‡]

インターネットの発達により、個人で簡単にインターネット放送を行える環境が整っている。個人でインターネット放送をしているとき、放送者が次に何を話せばいいかわからなくなるという問題が発生することがある。この問題は“ある話題の中で次に話す内容を忘れた”、“ある話題の終了後、次の話題が思い浮かばない”、“視聴者との話と関連した話題が思い浮かばない”の3つの場面に分けられる。本研究は、3つの場面において、放送者が次に何を話せばいいのかわからなくなるという問題を解決することを目的とする。ここで、話題を収集し、各場面に応じて適切に選択および提示することで、放送者が放送を続けるための支援を行う手法を提案する。また、提案手法を実現したシステムである“言うストリーム”を実装し、提案システムを用いて実際にインターネット放送を行ったところ、提案システムが放送者を支援するユースケースを得た。これは、提案手法は放送者が放送を続けるための支援ができることを示唆している。

IUstream: Personal Live Streaming Support System with Real-Time Topic Recommendation

Soya Kirito[†] Keiko Yamamoto[†] Itaru Kuramoto[†]
Yoshihiro Tsujino[†] Mitsuru Minakuchi[‡]

It has been much easier to perform live streaming via the Internet. When some performers broadcast, it is occurred that the performers have no idea for what they should talk next. Such cases are classified into three: the performers forget a phrase which they want to talk next in a topic, the performers cannot remind a new topic after finishing a certain topic, and the performers cannot realize a topic that relates to some talk with audiences. IUstream is a personal live streaming support system which solves the problem by recommending proper topic to the performers in those cases, so that it supports them to keep talking in their video streaming. We found the cases that IUstream supports the performers in actual personal live streaming. This indicates that IUstream can support performers to keep talking in their webcast.

1. はじめに

インターネットの発達により、個人による情報発信がより身近なものとなった。ブログや動画投稿など、情報発信の手段は様々であるが、近年ではUstream[1]やニコニコ生放送[2]などのライブ配信サイトを利用することで、インターネット放送による情報発信が個人でも簡単にできるようになった。

齊藤[3]はインターネット放送の魅力として、その双方向性通信と個人による情報発信を挙げている。例えばニコニコ生放送では視聴者は放送に対するコメントをつけることができるので、放送者は自分が発信する情報について、視聴者とリアルタイムにコミュニケーションをとることができる。これにより、放送者は自身の近況を視聴者に報告し、それに対する視聴者からのコメントを得ることができたり、逆に視聴者からの質問に放送者が答えたりできる。

一般のテレビやラジオの番組では、放送する内容を原稿として時系列に沿ってまとめ、キャスターやDJ等の放送者が逐一参照できるようにしている。しかし個人でのインターネット放送では、放送のための原稿を用意しないことが多い。これは個人の放送のために毎回原稿を用意することが手間であるからだと考えられる。また放送者が視聴者とのリアルタイムなコミュニケーションを目的として放送を行っている場合、放送で話される話題は視聴者のコメントによって変化する。そのため話題が時系列に沿って書かれているような従来型の原稿では、その時点で放送者が話すべき話題が原稿にあったとしても、放送者はその話題を直ぐに参照することができない。また、原稿を参照せずに放送を続けたとしても、放送者は放送で話そうと思っていた話題を完全には覚えていないことがある。このため、次に何を話せばいいのかわからなかったり話し忘れてしまうという問題が発生することがあると考えられる。

本研究では、この「次に何を話せばいいのかわからなくなる」という問題を解決し、放送者が放送を続けられるように支援することを目的とする。

話すべき話題を提示する研究としては、岡田ら[4]による司会進行を支援するウェアラブルMCシステムが上げられる。MCシステムでは、その一機能である台本のトラック機能により、司会者が台本の中の次に話すべき部分を忘れてしまったときに、読むべき場所を提示することで司会進行を支援している。MCシステムは、ひとつの話題の中で次に話す内容を忘れてしまった場面は解決しているが、あらかじめ決められた話題のみを話すことを前提としているため、視聴者との話に関連した新しい話題を提示することはできない。そのため話題が変化する生放送に使うことはできない。

[†]京都工芸繊維大学
Kyoto Institute of Technology

[‡]京都産業大学
Kyoto Sangyo University

2. 放送支援手法

次に何を話せばいいのかわからなくなる状況は、以下の3つの場面に分けられる。

- (1) ある話題の中で次に話す内容を忘れた
- (2) ある話題の終了後、次の話題が思い浮かばない
- (3) 視聴者との話と関連した話題が思い浮かばない

ここで「話題」とはその場で話されている話の主題を指し、「内容」とは話を構成する要素である。これらの場面においては、それぞれ各場面に応じた適切な内容を放送者が参照できることが望ましい。放送者がある話題の中で次に話す内容を忘れた場面では、放送者が今話している話題の、放送者が話した直後の内容を参照できればよい。ある話題が終わったあと次の話題が思い浮かばない場面では、放送者が今まで話していた話題と異なる話題を参照できればよい。また、視聴者との話と関連した話題が思い浮かばない場面では、放送者が持っている視聴者との話と関連した話題を参照することができればよい。以上をまとめると、場面と提示内容の対応関係は表1のようになる。

以降、各場面に応じた適切な提示内容を決定するための手法について述べる。手法の全体像を図1に示す。

2.1 放送で話される話題の収集

放送で話される話題は、放送者個人が持つ話題か、視聴者から提供された話題のいずれかである。ブログは個人の情報発信ツールのひとつであるため、放送者自身のブログには放送者が伝えたい内容、つまり放送で話題となりうる内容が多く記述されていることが期待される。また、ニュースは新しい情報を手に入れる主要な情報源であるため、放送者が最近見たニュースには放送者が話したいと思う話題が含まれていると考えられる。そこで本手法では、放送者自身のブログと放送者が閲覧したWeb上のニュースからその記事を自動的に収集し、データベースに保存する。またこのとき、収集された記事の特徴量を計算する。これらは、2.2で述べる「放送者が話している内容の認識」をする際に用いられる。

2.2 放送者が話している内容に応じた提示内容の決定

(1) 放送者が話している内容の認識

放送者が2.1で収集した話題を話していれば、話題に含まれる文は放送者の発言と高確率で対応すると考えられる。そこで、放送者の発言中の語と収集した話題に含まれる語とを比較し、放送者の発言が、どの話題のどの文にあたるか（あるいはどれに

表 1 場面と提示内容の対応関係

Table 1 Suitable recommendation for the case of stopping speech.

場面	提示内容
ある話題の中で次に話す内容を忘れた	(a) 放送者が今話している話題の 放送者が話した直後の内容
次の話題が思い浮かばない	(b) 放送者が今まで話していた 話題と異なる話題
視聴者との話と 関連した話題が思い浮かばない	(c) 放送者が持っている 視聴者との話と関連した話題

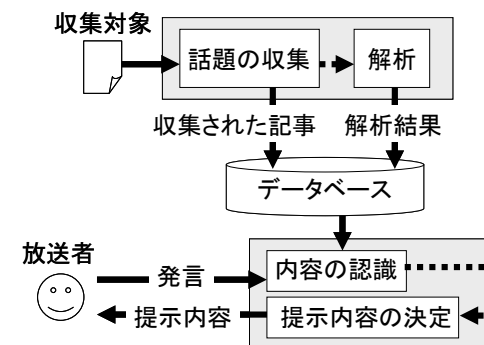


図 1 適切な提示内容を決定する手法の全体像

Figure 1 The structure of the proposed method.

もあたらぬか)を認識する。

(2) 認識結果に応じた提示内容の決定

放送者の発言から認識された文と現在提示している話題に応じて、以下の提示を行う。

まず、放送者の発言が現在提示している話題中の文であると認識されたときは、ある話題の中で話された内容とまだ話されていない内容を放送者の区別がつくように提示することにより、表1(a)に対応する内容を提示できると考えられる。そこで本手法では、放送者が既に話したと認識されている文を既読文、まだ話したと認識されていない文を未読文とし、既読文と未読文の区別がつくように話題を提示する。

次に、ひとつの話題の内容すなわち話題中の文が全て話されたとき、放送者には次に話す話題を提示する必要がある。このときに提示する話題の選択について、放送者にとっては、新しい話題のほうが提示されたときその内容を想起しやすいと考えられ

る。従って提示されるべき話題は、まだ提示していない話題のうち、より新しいものであることが望ましい。本手法では、ひとつの記事中の全ての文が認識されたときに、今まで提示していない、最も新しく収集された記事を提示する。これは、表 1(b)を提示することに対応する。

最後に、放送者の発言が現在話している話題と別の話題中の文であると認識されたときは、その認識された話題を提示するのが望ましい。これは、表 1(c)を提示することに対応する。ただ放送者には、今視聴者としている話に「合わせたい」「合わせたくない」という2つの傾向があると考えられる。従って、現在提示している話題か、視聴者との話と関連した話題か、どちらをより提示してほしいかという傾向は、放送者の好みや状況によって変化すると考えられる。そこで本手法では、どちらの話題をより提示してほしいかの傾向を放送者に入力させ、放送者が話している内容の認識に反映させる。

(3) 次に発言すべき文の強調表示

(2)で決定された各場面に応じた適切な話題を放送者に提示するとき、話題の中で次に発言する可能性の高い文を強調して提示することにより、放送者の次の発言を支援する。本手法では、強調表示される文を図 2 に示すフローチャートによって決定する。

3. 言うストリーム

2.で述べた手法を実現した提案システムである「言うストリーム」の実装について述べる。提案システムは、2.1で述べた手法を実現する記事入力・解析部と、2.2で述べた手法を実現する発話認識・提示部の2つに分かれる。本システムでは、放送者が話すであろう話題の集積を myDB と呼ぶ。

記事入力・解析部は、放送者がブログに記事を書き込んだり、ニュースサイトの記事を閲覧したりしたとき、それらの記事を自動的に myDB に登録する。そして、放送者が現在何を話しているのかを認識するために必要な情報を得るために、集積された記事を解析し、解析した情報を myDB に追加する。

発話認識・提示部では、解析された情報と放送者の発言から、放送者が現在どの話題のどの文を話しているのかを認識する。本研究では放送者が現在話しているとシステムが認識した文のことを認識文と呼ぶ。そして認識文に応じて、myDB に集積された記事を適切な提示手法を用いて提示する。さらに提示する記事に含まれる特定の文を強調表示する。

以降、記事入力・解析部について 3.1 で、発話認識・提示部について 3.2 でそれぞれ詳しく述べる。

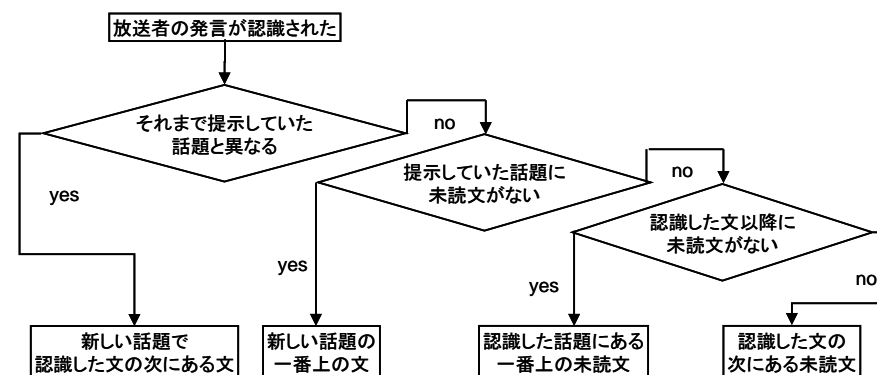


図 2 強調表示をする文の決定

Figure 2 Determination of the sentence for emphasizing.

3.1 記事入力・解析部

記事入力・解析部では記事を取得、解析し、myDB を作成する。

記事入力部では、放送者が普段書き込みをしているブログ・閲覧しているニュースサイトから、記事を自動で取得する。取得された記事はその順番でmyDB上に登録される。このときにそれぞれの記事に対して順に記事番号と振る。この記事入力部を Mozilla Firefox用のアドオンとして実装した。また、対象とするブログシステムをソーシャル・ネットワーキング サービスmixi (ミクシィ) [5]とした。これに放送者自身のブログの閲覧・書き込みを行うと、ブログのタイトルとブログの本文が、それぞれ記事タイトルと記事として取得される。

記事解析部では、まず入力された記事を文に区切る。このとき、文の終わりの記号になりやすい「。」「?」および改行文字を文の終わりとし、この記号の後ろで区切る。

次に、各文に対して形態素解析を行い、各文に含まれる自立語を抽出する。以下、記事から抽出された自立語のことをキーワードと呼ぶ。形態素解析ツールには MeCab[6]を用いた。

最後に、tfidf法を用いて各記事および記事中の各文において特徴的なキーワードを求める。tfidf法は主に重要語に注目した重要文抽出や、文書の要約などに使われている。以降、各記事における特徴的なキーワードの評価値として以下の式により定義される tfidf 値を用いる。

$tf(t, d)$ はある記事 d におけるキーワード t の出現頻度であり、 $n(t,d)$ を記事 d におけるキーワード t の出現回数としたときに、

$$tf(t, d) = \frac{n(t, d)}{\sum_{k \in d} n(k, d)} \quad (1)$$

と定義される。また $idf(t)$ は全ての記事におけるキーワード t の逆出現頻度であり、

$$idf(t) = \log \frac{|D|}{|\{d : t \in d\}|} \quad (2)$$

と定義される。D は myDB に含まれる記事全体を指し、 $d \in D$ である。よって分子は myDB に含まれる記事の総数を示し、分母は myDB に含まれる全ての記事のうち、キーワード t が含まれる記事 d の数を示す。

このとき $tf-idf$ 値を

$$tfidf(t, d) = tf(t, d) \cdot idf(t) \quad (3)$$

の式で求める。

さらに、記事中の各文における特徴的なキーワードを求める。本研究では、各文間に含まれるキーワードを特徴づける値として $tfisf$ 値(Term Frequency - Inverse Sentence Frequency)を $tfidf$ 値同様に、以下の式により定義する。

$tf(t, s)$ はある記事 d 中の文 s におけるキーワード t の出現頻度であり、 $n(t, s)$ を文 s におけるキーワード t の出現回数としたときに、

$$tf(t, s) = \frac{n(t, s)}{\sum_{k \in s} n(k, s)} \quad (4)$$

と定義される。また $isf_d(t)$ は全ての文におけるキーワード t の逆出現頻度であり、

$$isf_d(t) = \log \frac{|d|}{|\{s : t \in s\}|} \quad (5)$$

と定義される。ここでは、ある記事 d を記事中に含まれる文の集合と考えるので、分子は記事 d に含まれる文の総数を示し、分母は記事 d に含まれる全ての文のうち、キーワード t が含まれる文 s の数を示す。

このとき $tfisf$ 値を

$$tfisf_d(t, s) = tf(t, s) \cdot isf_d(t) \quad (6)$$

の式で求める。

この方法で求めた $tfidf$ 値と $tfisf$ 値を、取得した記事とともに myDB に登録する。

3.2 発話認識・提示部

発話認識・提示部は放送者の発言を入力として認識文を決定する認識部と、認識文に応じて適切な発話候補文を提示する提示部に分かれる。本システムでは、放送者の音声認識のためのソフトウェアとして Julius[7]を用いた。

発話認識・提示部ではまず、Julius により認識された放送者の音声が発言として入力される。そして放送者の発言から自立語を抽出し、抽出された全ての自立語を認識部への入力とする。認識部ではこの入力に基づいて認識文を決定し、提示部ではこの認識文に応じて適切な提示手法を用いて発話候補文を提示する。

以降、放送者の発言時に提示部で提示されている記事を現在記事、放送者の発言から抽出された自立語を発話キーワードという。

3.2.1 認識部

認識部は発話キーワードを入力とし、それに基づいて認識文を最尤推定により決定する。ここで、myDB に含まれる全ての文に対して定まる文発話値を定義する。文発話値は、ある文に発話キーワードが含まれていたとき、そのキーワードの $tfidf$ 値、 $tfisf$ 値に応じて増加する値である。文発話値は発話キーワードが入力されるごとに更新される。いずれかの文発話値がある閾値を超えたとき、分発話値と閾値との差が最も大きい文が認識文として決定される。

アルゴリズムを以下に示す。

myDB に含まれる全ての文 s の文発話値 $v(s)$ について、 $v(s) \leftarrow 0$

myDB に含まれる全ての記事 d について

記事 d に含まれる全ての文 s について

全ての発話キーワード k について

d が現在記事のとき

$k \in s$ ならば、 $v(s) \leftarrow v(s) + tfisf_d(k, s)$

d が現在記事ではないとき

$k \in s$ ならば

$$w(p, d, k) \leftarrow \frac{1}{2} p \left(e^{-num(d) \cdot C_s} + \frac{tfidf(k, d)}{\max_i \{tfidf(t, d)\}} \right) \quad (7)$$

$v(s) \leftarrow v(s) + tfisf_d(k, s) w(p, d, k)$

$L(s) \leftarrow C_s \sum_{t \in s} tfisf_d(t, s)$

$v(s) > L(s)$ ならば、 $M(s) \leftarrow v(s) - L(s)$

$M(s)$ が最大となる文を認識文とする

式(7)の $w(p,d,k)$ は、現在提示している話題か、視聴者との話と関連した話題か、どちらをより提示してほしいかという放送者の傾向を文発話値の増加量に反映させるためのものである。 p は放送者の傾向を示す値であり、発話キーワードとは別に入力される。 $num(d)$ は d の記事番号、 C_g は記事番号が古い記事の文発話値の増加量を抑えるための定数である。括弧内第2項は重要ではない k の入力に対する文発話値の増加量を減らすための、 $tfidf(k,d)$ と記事 d 内での最大の $tf-idf$ 値との割合を示す値である。括弧内の各項の値は 0~1 までの実数値となり、それを平均したものに p を掛けることで、放送者の傾向を反映させた重み付けを行う。

C_g と C_s はシステム内部で設定するハイパーパラメータである。 C_s の値を大きくすると、文が認識される回数が増え、それだけ放送者の想定外の文も認識される可能性が高くなる。本研究では、被験者自身が書いた日記の内容について実験者に話させる実験を行い、被験者の話と日記を比較することで $C_g \cdot C_s$ の最適値を求めた。

3.2.2 提示部

提示部では 3.2.1 で認識された結果に応じて、放送者に提示する記事と強調する文を決定し、提示する。

放送者に対する提示画面を図3に示す。提示画面の左側には、システムに登録されている話題の一覧が表示される(図3①)。話題を選択することで、対応する記事を表示することができる。提示画面の上部には、現在提示されている話題の提示画面が表示される(図3②)。記事はその下に表示され、既読分は灰色で(図3③)、未読文は黒で(図3④)、強調して提示される文は背景が黄色で表示される(図3⑤)。また、「飛ばしたところへ」ボタン(図3⑥)を押すと、表示している記事の一番上にある未読文が、「現在位置」ボタン(図3⑦)を押すと、提案システムにより提示された文が表示される。また、右下のスライダ(図3⑧)で式(7)中の p の値を設定でき、今話している話題と関連する話題か、今提示している話題かどちらをより提示しやすくするかを調節することができる。

放送者は最も新しい話題を最初に話す可能性が高い。そこで提案システムが起動した直後は、myDBに登録されている記事の中で最新の記事を提示する。

4. 「言うストリーム」のユースケース

2.に示した問題が起きたときに、「言うストリーム」が放送者を支援することができるのかを調べるために、実際にシステムを用いた生放送を行い、システムのユースケースを調査した。この際に用いるインターネット放送サービスとして、ニコニコ生放送を選択した。ニコニコ生放送を選択したのは、視聴者が多くコメントが匿名であり、

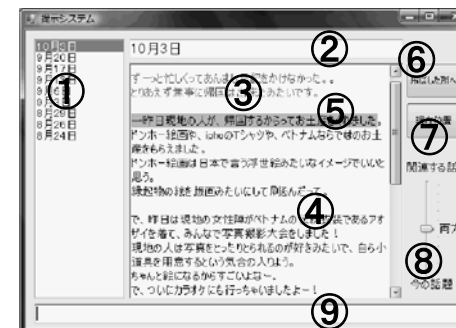


図3 「言うストリーム」の提示画面
 Figure 3 Appearance of IUstream.

Ustream等の個人生放送サイトと比較して視聴者がコメントをする数居が低いためである。従って、放送を行う上で視聴者とのコミュニケーションがとりやすいと考えられる。

「言うストリーム」を用いて生放送を行っているときの画面を図4に示す。計3回90分の放送のうち、関連する話題が思い浮かばなかった場面が4回あり、その中でシステムが新しい話題を提示することで放送者を支援した場面が2回、話題の提示ができずに放送者を支援できなかったときに2回あった。また、放送者が特定の固有名詞を忘れていたときにシステムを参照することで思い出すことができた場面が1回あった。

4.1 放送者を支援した例

「言うストリーム」が新しい話題を提示した例として、放送者が放送の中で最近の映画の話をしていた場面が挙げられる。この例では、放送者が当時話題に上がっていた映画名を挙げたときに、システムがその映画名と映画というキーワードから、映画会社の社員の人と会った話を提示した。放送者は最近の映画の話に関連する話題として社員の人であった話を思い浮かべることができなかったが、システムが関連する話題を提示したことで、うまく関連する次の話題につなげることができた。

もうひとつの例では、視聴者からある話題を提供されたとき、放送者はその話題がシステムに登録されていることを知っていた。このときは放送者がこの話題について視聴者に返答したことでシステムがキーワードを認識し、放送者に適切な話題を提供することができた。このとき、放送者は視聴者が提供していた話題がシステムに登録されていることを知っていたので、システムのインタフェースを操作することで適切な話題を提示させることが可能ではあった。しかしこの例では、システムが放送者の



図 4 言うストリームを使用して生放送を行っているときの画面
Figure 4 Live streaming with IUstream.

発言を認識することで、システムは放送者によるインタフェースの操作無しに適切な話題を提示することができた。

また、放送者が特定の固有名詞を忘れていたときに「言うストリーム」を参照することで思い出すことができた例として、旅行で行った観光名所の名前を思い出すことができなかった場面が挙げられる。放送者は放送中に名前を思い出すことができなかったが、ブログにその名前を書いたことは思い出したため、システムを操作して該当する日記を見つけることで、観光名所の名前を思い出すことができた。

4.2 放送者を支援できなかった例

「言うストリーム」が関連する話題を提示できなかった例として、旅行先での驚いた出来事についての話題をシステムが提示できなかった場面があった。これは、放送者が視聴者から旅行先でカルチャーショックだった出来事について聞かれたときに、システムは放送者の発言から「驚いた」「ショック」のキーワードを認識した。このとき、旅行中にトラブルがあり驚いた出来事についての話題がシステムに登録されていたのにもかかわらず、システムは該当する話題を提示できなかった。これは驚いた出来事の話が書かれている記事に「驚いた」「ショック」のキーワードが入っていないことが原因である。本手法では今話している内容の認識に単純なキーワードマッチングを用いているため、例えば表記のゆれなどにより、記事に放送者が入力したキーワードが書かれていないときは関連する話題を認識させることができない。このことはもうひとつの例でも同様であった。

また、今回は日記のみから放送で話す話題を収集したが、約2週間分の日記の登録をしたのみで、3回分の放送時に提示される話題の数は十分であった。しかし放送の

頻度を上げるのであれば、より多くの話題を収集する必要があると考えられる。話題の収集対象をブログ以内にも拡大することは、放送者が話すことができる話題の増加につなげることができると考えられる。特に Twitter に代表されるマイクロブログはブログよりも手軽に書き込むことができるため、使用者が人に話したい話題を書き込む頻度も高いと考えられる。この話題の収集対象の拡大は、今後の課題として挙げられる。

5. まとめ

本研究では、次に何を話せばいいのかわからなくなるという問題を解決し、放送者が放送を続けられるように支援することを目的とし、各場面に応じた適切な話題を提示する「言うストリーム」を実装した。

システムを用いて実際にインターネット放送をしたところ、システムが放送者を支援するユースケースを得た。これは、提案手法で放送者が放送を続けるための支援ができることを示唆している。今後の課題としては、システムのユーザビリティに関する評価と話題の収集対象の拡大が挙げられる。

参考文献

- 1) USTREAM, You're On. Free LIVE VIDEO Streaming, Online Broadcasts. Create webcasts, video chat, stream videos on the Internet. Live streaming videos, TV shows (オンライン), 入手先 (<http://www.ustream.tv/>) (参照 2011-2-28)
- 2) ニコニコ生放送 (オンライン), 入手先 (<http://live.nicovideo.jp/>) (参照 2011-2-28)
- 3) 齊藤 義仰: インターネット放送の魅力と可能性, 情報処理 51(1), pp.64-67, (2010)
- 4) 岡田 智成, 山本 哲也, 寺田 努, 塚本 昌彦: 司会進行を支援するウェアラブル MC システムの設計と実装, 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 (WISS 2009) 論文集, pp.35-40, (2009)
- 5) ソーシャル・ネットワークワーキング サービス [mixi(ミクシィ)] (オンライン), 入手先 (<http://mixi.jp/>) (参照 2011-2-28)
- 6) MeCab 「MeCab: Yet Another Part-of-Speech and Morphological Analyzer」
<http://mecab.sourceforge.net/>
- 7) Julius 「大語彙連続音声認識エンジン Julius」
<http://julius.sourceforge.jp/>