

帰納的学習を用いた議事録自動生成手法の性能評価

阿部 里美 荒木 健治
北海道大学大学院 情報科学研究科

本稿では、議事録の自動生成についての研究を報告する。会議の会話を対象とした自動要約は現在研究段階にあり、人手により作成したルールを用いた研究がいくつか行われている。しかし、会議での会話のように、書き言葉の表現に比べて細かな違いが現れやすい話し言葉では、規則を人手によって与える場合、全てを網羅することは困難である。そこで、本稿では発話の表現などの特徴からなる議事録自動生成ルールを、帰納的学習により自動的に獲得し、議事録の自動生成を行う手法を提案する。評価実験を行った結果、ベースラインとして定めた tf-based 手法と比較し、正解データとのコサイン距離において良好な結果を得たことから、本手法の有効性が確認された。

Evaluation of Automatic Generation Method for Summarization of Meeting Recordings Using Inductive Learning

Satomi Abe Kenji Araki
Graduate School of Information Science and Technology, Hokkaido University

In this paper we propose a method for automatic generation of meeting summaries. Automatic summarization of meeting recordings is being developed, and there are some researches that use rules given manually. But conversation at a meeting is in spoken language, which has much variety of expressions in comparison with writing language. It is difficult to generate all the rules manually. Therefore we created a method for automatic generation of meeting summaries, which acquires the rules containing speech phrase patterns and so on by using Inductive Learning. We performed experiments to evaluate the performance of our system and showed that our method is better than tf-based baseline method, on cosine distance with human generated summaries.

1. はじめに

インターネットの普及等により、必要な情報を絞り込むための技術として自動要約技術の研究が盛んである。近年その対象は、複数文書の要約や、対話の要約等、様々に広がりつつある[1]。本稿では、会議の書き起こしを対象とした自動要約、すなわち議事録の自動生成についての研究を報告する。

会議の会話を対象とした研究は、現在研究段階にあり、会議コーパスから重要文抽出を行う研究[2]や、チャット会議ログから重要箇所抽出を行う研究[3]、国会会議録を対象として文内短縮を行う研究[4]などが行われている。文献[2]では、MMR や LSA などによる重要文抽出の結果の比較を行っているが、いずれの手法も単語の出現頻度からなる単語重要度を用いており、議長の議事進行の発話など、会議会話独特の特徴は利用していない。我々は重要語の利用に加えて、このような特徴が重要部分の判断に有効であると考えている。また、文献[3]、文献[4]は、いずれも重要文抽出や換言・削除のルールを人手により作成している。会議での会話は、通常の話し言葉に

比べ、書き言葉に近い発話が行われるが、新聞記事などの文章と比較して、細かな表現の違いや、発話者による特徴の違いなどが現れる。人手によるルール作成を行う場合、そのような細かい違いを考慮したルールを全て与えるのは困難であり、網羅するためには対象とするデータに応じて、足りないルールを追加していくことが必要となる。

そこで、本研究では、議事録自動生成ルールを帰納的学習[5]により逐次的に獲得し、それを用いて議事録の自動生成を行うというアプローチをとった。ここで言う帰納的学習とは、我々が提案した学習手法であり、実例からそこに内在する規則を再帰的に自動獲得する学習手法のことを指す。帰納的学習は解析的知識を人手により与える手法とは異なり、比較的少量の用例からでも多くのルールを獲得することができるという特徴がある。また、実例からルールを獲得していくため、入力されるデータに適応した、より適切なルールを獲得することが可能である。

また、現在、会議会話の要約に関する研究のほとんどが、ほぼ抜き出した文そのままに近い形で出力しており、実際に人間が作成する議事録とは大きく

異なるものである。そこで、本稿で提案するシステムは、抽出した重要文の文体を変換し、決定事項や意見等、文の表す内容によって整理し、より議事録の形に近づけた状態での出力を目指す。

2. システムの概要

概要を図1に示す。本システムはまず会議の会話履歴から議事録に掲載すべき情報を含む文（以降重要文と呼ぶ）を抽出する。続いて、抽出した文の文体を議事録に合致した形に変換し、更に、文内短縮処理を施してより簡潔な文を生成する。その後、文を並べ替え、整理して議事録を出力する。議事録の出力が終わると、各部で使用するルールを学習を行う。最後に、出力された議事録の正誤を元に、使用したルールの正誤を判断し、フィードバックする。

本章では、各処理部の概要について述べる。

2.1. 重要文抽出部

重要文抽出部では入力として、形態素解析(茶釜[6]を使用)により品詞が付加された会議会話の書き起こし文を受け取る。受け取った文書に対して、重要文抽出ルールを用いて各文の重要度を計算し、重要度の大きな文を重要文として抽出する。

続いて、学習用の正解データ(以降正解重要文と呼ぶ)と既に獲得済みのルールとの組から、共通部分及び差異部分を比較することで、より一般化された新たなルールを獲得する。最後に、正しい重要文を抽出できたルールの尤度を増大させ、誤った文を抽出したルールの尤度を減少させることにより、抽出に使用したルールの尤度を更新する。

以上のように、会議書き起こしデータを新しく入力する度に、それまでに獲得したルールを用いた重要文の抽出、入力されたデータからのルールの学習、抽出した文の正誤を元に、使用したルールの正誤をフィードバック、という順に処理を繰り返すことで、システムは逐次的に学習を進め、性能が向上していくことになる。

以下ではシステムが使用する重要文抽出ルール、及び各処理の概要について述べる。詳細については先行研究である文献[7]に示す。

2.1.1. 重要文抽出ルール

重要文抽出には、次の2種類のルールを用いる。

素性ルール

正解重要文に対して、文の長さ(単語の数)、全文中での出現位置、一つの発言中での出現位置、単語のtf*idfの和、話者が誰か、という5つの素性を、それぞれ0, 1, 2, ...といった値で数値化する。た

例えば、全文中での出現位置という素性は、全文中での出現位置が序盤であれば0、中盤であれば1、終盤であれば2となり、話者が誰かという素性については、話者が議長であれば1、それ以外であれば0となる。これら5つの数字を並べたものを文の状態パラメータとし、これを素性ルールとして用いる。具体的な数値の割り当てについては先行研究である文献[8]に記載しているとおりである。

パターンルール

「@で/助詞 よろしい/形容詞 です/助動詞 か/助詞 . /記号」といった表現のパターンからなるルールである(@は任意の文字列に対応する変数)。

2.1.2. 学習処理部

学習処理部では、正解重要文から2.1.1で述べた2種類のルールを重要文抽出ルールとして獲得する。獲得方法を以下に示す。

素性ルールの獲得

正解重要文に対し、2.1.1で示した文の状態パラメータを求め、それを素性ルールとして獲得する。

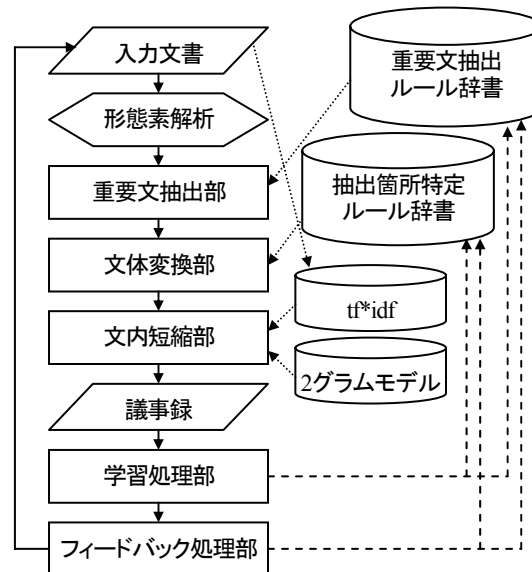


図1: システムの概要

- (a) 正解重要文
それでは、この報告書を提出ということにしたいと
 思います。
- (b) 獲得済みルール
それでは、@するということにしたいと
 思います。
- (c) 新しく生成されたルール
 それでは、@ということにしたいと
 思います。
 ※品詞は省略
 ※@は任意の単語列を表す変数

図2: パターンルールの獲得例

パターンルールの獲得

正解重要文そのもの及び既に獲得したルールと正解重要文との組から、字面・品詞共に共通する部分を残し、異なっている部分を変数に置き換えたものを、一般化された新たなルールとして獲得する。

図2に獲得例を示す。図2において、正解重要文(a)と既に獲得済みのルール(b)の組の場合、共通部分である下線部を残し、その他の部分を変数に置き換えたルール(c)が生成される。新しく生成したルール、及び正解重要文そのものをルールとして獲得する。

議事録に必要な文は、決定事項を表す文、会議中に出た意見を表す文など、その性質によって特徴が異なると考えられる。そこで本手法では、正解データである各正解重要文には、表1に示す8種類のカテゴリのうちいずれに属する文か、という情報を付加している。ルールの生成元である正解重要文のカテゴリが、新しく生成したルールの属するカテゴリとなり、ルールはカテゴリ毎に獲得される。カテゴリ分けをすることで、より適切なルールの生成、重要文の抽出が行われる。

2.1.3. 抽出処理部

重要文の抽出は文の重要度を元に行う。重要度はその文にルールを適用した際のスコアを元に計算する。スコア計算式は文献[7]で示したものと同様のものをを用いるため、詳細を省略し、概要のみを述べる。

素性ルールのスコア

2.1.1で述べた対象文の素性と、各素性ルールの素性を各々比較し、どの素性が一致しているのか、その素性は他の素性に比べてどの程度重要か、を元に求められる。また、そのルールが適用されたとき及びルールが獲得された時にカウントされる値(出現回数)、そのルールを適用して正しい重要文が抽出された時にカウントされる値(正適用回数)、誤った文が抽出された時にカウントされる値(誤適用回数)から求められる、各ルールの尤度も考慮される。

パターンルールのスコア

対象文にマッチしたルールについて、素性ルールの尤度と同様にして求めた尤度をそのルールのスコアとする。

以上のようにして、対象文に対する全てのルールのスコアを求める。そして閾値を越えたルールについて、カテゴリ毎に、パターンルールのスコアの和と素性ルールのスコアの和の重み付き和を求め、最も大きな値となったカテゴリのルールを適用し、そのスコアを文の重要度とする。重要度が閾値を超えた場合、重要文として抽出する。

2.1.4. フィードバック処理部

システムが抽出した文が正しいか否かを人手で判断し、その正誤に応じて、2.1.3で述べた出現回数、正適用回数、誤適用回数を調整することにより、システムにフィードバックをかける。

2.2. 文体変換部

文体変換部では、2.1の重要文抽出によって抽出された文を、議事録として適切な文体に変換する。ここでは、学習処理、抽出箇所特定処理の概要について述べ、詳細については文献[8]に示す。

システムが抽出した文は、その文の抽出に用いたルールのカテゴリに属するものと見なすことができる。そこでそのカテゴリ情報を利用し、表1に示すテンプレートの対応表において、“~”に入れるべき箇所を、抽出した文の中から特定し、その箇所をテンプレートに当てはめることで文体変換を行う。

処理の流れは、まず抽出箇所特定ルールを用いて、文中のどの部分を抜き出すべきかを特定する。抽出箇所特定ルールは、テンプレートへ抜き出すべき箇所の前後3単語からなる。3単語としたのは、前後2~6単語まで単語数を変更して実験を行った結果、最も良好な結果を示したためである。ルールは『ませ/助動詞 ん/助詞 . /記号 ~ と/助詞 言う/動詞 こと/名詞』のような形で表され、この場合、『ませ/助動詞 ん/助詞 . /記号』と『と/助詞 言う/動詞 こと/名詞』に挟まれる部分が抜き出すべきである、ということを表している。抽出箇所特定後、テンプレートに当てはめ、文体変換を行う。その後、図1に示す学習処理部において、正解重要文と正解変換文の組からどのような箇所を抽出すべきかということの内容語の一致から判断し、新たなルールを獲得する。更に、新しく獲得したルールと既に獲得済みのルールとの組から共通部分、差異部分を元に、より一般化されたルールを獲得する。獲得例を図3に示す。最後に2.1の重要文抽出と同様、フィードバック処理により、正しい箇所を抜き出すことができたルールの尤度を増大させ、誤った箇所を抜き出して

表1: カテゴリとテンプレートの対応

カテゴリ	テンプレート
前回までの話題	(そのまま表示)
報告	~について~から報告があった。
説明	~について~から説明があった。
審議内容	~について審議した。
確認内容	~について確認した。
決定事項	~こととなった。
意見	(そのまま表示)
今後の予定	~こととなった。

- | |
|---|
| <p>(a) 今回獲得したルール
ませ/auxv n/p . /s ~ と/p 言う/v こと/n</p> <p>(b) 既に獲得済みのルール
, /s それでは/conj , /s ~ と/p 言う/v こと/n</p> <p>(c) 生成されるルール
@1/@2 @3/@4 @5/s ~ と/p 言う/v こと/n</p> <p>※“@1”等は変数を表す</p> <p>※接続詞=conj, 助詞=p, 記号=s, 助動詞=auxv,
動詞=v, 名詞=n とする</p> |
|---|

図3: 抽出箇所特定ルールの獲得例

しまったルールの尤度を減少させることで、ルールの尤度を更新し、フィードバックをかける。

2.3. 文内短縮部

文体変換を施した後、その文を文内短縮することによって、不要な部分を除き、より簡潔な文を生成する。文内短縮により生成される文は、元の文の持つ重要な情報を含み、文としての自然さを保持していなければならない。そこで本手法では、重要語の判断に $tf*idf$ を、文としての自然さを保つために $2gram$ を用いて文内短縮を行う。また、より自然な文を生成するため、単語を削除する際には係り受け関係を考慮した処理を行う。

具体的な手順を以下に示す。

1. 文を文節で区切り、係り受けを解析する (CaboCha[9])。
2. 各文節と、その文節に係る文節に含まれる単語の $tf*idf$ の平均値を求める。この係り受けを考慮した塊(以降節と呼ぶ)を、削除単位とする。
3. 2.で求めた $tf*idf$ が小さい節3つを削除候補とする。
4. 3.で得られた削除候補のうち、その節を削除した場合の文のパープレキシティが、最も小さいものを、その節を削除すると、より日本語らしい文になると判断し、削除を行う。
5. 下記の終了条件を満たさなければ、2.~4. を繰り返す、削除を進める。
 - ・ 削除後の文の $tf*idf$ の合計が元の文の9割以下
 - ・ パープレキシティが元の文よりも大きい

以上の手順に従って文内短縮を施す。手順3において削除候補を3つとし、手順5において $tf*idf$ 合計が元の文の9割以下としたのは、様々な値で実験を行ったところ最も良好な結果を示したためである。文の自然さの判断には $2gram$ モデルから算出される単語パープレキシティを用いた。単語パープレキシティとは、文の生起確率を単語あたりに平均して逆数を取った値であり、値が低いほど日本語として出

現する可能性が高いことを表している。すなわち、日本語らしい文であると考えられることができる。

$tf*idf$ で用いる df (document frequency)には毎日新聞2000年、2001年の2年分のデータを用いた。

システムが抽出した全ての文の文体変換及び文内短縮が終了すると、それらを表1のカテゴリの順に並べ替えて整理し、議事録として出力する。

3. 評価実験

システムの性能を評価するため実験を行った。本手法では全てのルールを学習により獲得するため、予めルールを与える必要はない。そこで重要文抽出ルール及び文体変換の抽出箇所特定ルールは、全て空の状態から開始した。重要文抽出で用いる閾値等は様々な値で実験を行ったところ、最も良好な結果を示した値を用いた(重要度閾値=1.9, 素性ルール閾値=0.6, パターンルール閾値=0.5, 素性ルールに対するパターンルールの重み γ =0.4) また、重要文は1文のみでは情報が不足することが多いため、本手法では隣接する2文を単位として重要文の抽出を行う。

3.1. 実験データ

実験には、18回分の会議会話書き起こしデータ[10]を用いた。文数は1会議平均381文であり、全データの総文字数は376,686文字(1会議平均20,927文字)であった。また、平均時間は約1時間半であった。

学習に使用する正解重要文は、文献[10]において会話データと共に公開されている議事録を元に、対応する文を正解重要文として第一著者が定めた。ただし、広い範囲の文章をまとめたものや、会話中に直接述べられていない内容を含んだものは重要文を定義することができないため、今回は扱っていない。正解重要文の数は会議によって異なるが、1会議平均 4.6×2 文(2文単位で扱うため)である。また正解の文体変換文についても、会話データと共に公開されている議事録を元に、第一著者が定めた。

3.2. 評価方法

現在、議事録自動生成において確立された評価方法が存在しないため、評価方法、及び結果を比較するためのベースラインは、NTCIR2-TSC1[11]のタスク「人間の自由作成要約と比較可能な要約」生成におけるcontent basedな評価を参考にした。具体的な評価方法は、人間が作成した正解議事録と、システムが出力した議事録を形態素解析(茶筌[6])し、内容語のみを抽出する。そして、正解議事録とシステムが生成した議事録の単語頻度ベクトル間の距離を計算し、どの程度内容が単語ベースで類似しているかを求めるものである。ここでは、内容語として名詞、

動詞、形容詞、未定義語を使用し、ベクトルの要素は、各内容語の $tf*idf$ 値とした。ベースラインもNTCIR2-TSC1に習い、 tf -based手法を用いた。これは、各文について内容語の tf の和を計算し、それを文の重要度として重要文抽出を行うものである。

3.3. 実験結果

図4は重要文抽出のみの結果である。横軸は入力した会議の数を、縦軸は再現率と適合率から求められるF値を表している。学習が進むに従って、F値が徐々に上昇しており、より適切なルールが獲得され、性能が向上していく様子がわかる。最終3データではF値0.357を得た。一般的に議事録では、会話の内容がとても大きく圧縮される。実際に今回使用した実験データでは、最終3データにおける要約率が平均2.6%であった(重要文を2文単位で扱っているため、抽出文数*2/総文数によって要約率を求めた)。要約率が極めて小さいこと、及び tf -based手法によるF値がデータ全体の平均で0.042であったことを考慮すると、比較的良好な結果であると言える。しかし、グラフに示すとおり学習はまだ飽和していない。最終的にどこまで性能が向上するのかわかることを確認するためには、更に学習データを追加する必要がある。表2では、正解議事録とシステムが出力した議事録とのコサイン距離を示している。各手法において、上段は単語ベクトルの要素を $tf*idf$ とした場合、下段は出現頻度とした場合の結果である。提案手法については、抽出した文そのままの文、文体変換まで施した文、文内短縮まで施した文の3種類とのコサイン距離を求めた。ベースラインの抽出文数については、重要文の数は会議によって異なるため、提案手法では閾値により抽出文数を決定していることから、比較のためにベースライン手法での抽出文数もシステムが抽出した文数に合わせて実験を行った。また、提案手法は逐次学習を行うため、学習が始まったばかりの時点では、まだルールがあまり獲得されていない。このため学習の進んだ状態である、最終6会議文の出力結果を用いて比較を行った。最後に、実際にシステムが生成した文の例を図5に示す。図5において、「元の文」の下線部は、次の処理である文体変換処理によって抜き出された箇所であり、「文体変換後の文」の下線部は文内短縮処理によって削除された部分である。文内短縮によって平均約20%短縮することができた。

3.4. 考察

表2において、文内短縮まで行った場合の提案手法のコサイン距離と、ベースライン手法のコサイン距離の値を比較すると、 $tf*idf$ をベクトル要素とした

場合、提案手法がベースラインを上回っており、単語ベースでの類似度から判断して、人間が作成した正解議事録により近い文を生成できたと言える。抽出された重要文自体を比較してみても、最終6データにおける提案システムのF値は平均0.269であるが、ベースライン手法では同データにおいて平均0.005と極めて小さい。抽出した重要文にこれだけの差異があるのに対し、コサイン距離による違いが比較的小さいのは、ベースラインの tf -based手法は tf の大きな単語を多く含む文を抽出するため、誤って抽出した文中にも tf の大きな語が存在している可能性が高い。そして正解議事録にも誤った文中の tf の高い単語と同じ単語が含まれていた場合、文意は異なっても $tf*idf$ をベクトル要素とした類似度計算では類似度が高くなってしまふ。このことが原因でコサイン距離の差異が小さくなったと考えられる。そこで、単語ベクトルの要素として、重要と考えられる語に重みを与える $tf*idf$ ではなく、単純に単語の出現頻度を用いた場合のコサイン距離についても比較を行った。その結果が、表2の各手法の下段に示す値であり、提案手法がベースラインを大幅に上回る結果となった。このことから、提案手法の方が、より単語そのものの類似度が高いと言える。

次に、抽出したそのままの状態の文と、文体変換後の文についてコサイン距離を比較すると、ベクトル要素を $tf*idf$ 又は出現頻度どちらにした場合も、文体変換後の文が抽出文の結果を上回っている。文体変換処理により、重要な箇所のみを上手く抜き出し、テンプレートに当てはめ、より正解議事録に近い文を生成することができたと言える。また、文体変換

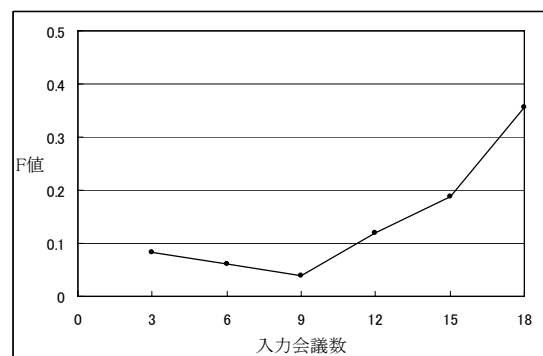


図4: 3 会議毎の F 値平均の推移

表2: 正解議事録とのコサイン距離の比較

	コサイン距離		
	抽出文	文体変換後	文内短縮後
提案手法	0.3742	0.4099	0.4108
	0.3077	0.3327	0.3270
base line (tf based)	0.3788		
	0.1937		

[元の文]

それでは、一通り来ましたが、次は総会になってしまいますので、事務局の方で整理して、総会以後の話を要約して、報告書の文案をつくらせていただいて、事前に配付させていただくような形でよろしいでしょうか。もう一回集まるのは大変だと思います。

[文体変換後の文]

一通り来ましたが、次は総会になってしまいますので、事務局の方で整理して、総会以後の話を要約して、報告書の文案をつくらせていただいて、事前に配付することとなった。

[文内短縮後の文]

来ましたが、総会になってしまいますので、事務局の方で整理して、話を要約して、文案をつくらせていただいて、配付することとなった。

[正解議事録中の文]

事務局で整理して報告書の文案をつくり、事前に配布することとなった。

図5: 文体変換, 文内短縮処理の結果

まで行った文と、さらに文内短縮まで行った文について比較を行うと、ベクトル要素に出現頻度を用いた場合には、若干の低下が見られるが、文内短縮処理によって、平均約20%の短縮を行ったことを考慮すると、比較的良好な短縮が行われたと考えられる。しかし、図5のように文内短縮処理後に、口語的な部分が残ってしまう例も見られた。また、本手法では $tf \cdot idf$ の平均を用いて削除する文節を決定しているが、図5の「報告書の」のような、人間が重要であると考えられる語でも、 $tf \cdot idf$ 値が低いために誤ってその部分を削除候補としてしまう例が見られた。削除過程を分析してみると、現在 idf の計算式は、 $\log(\text{総文書数}/df(\text{単語})+1)$ を用いているが、会議会話のように比較的文書が長い場合、「,」などの出現頻度 tf が、 idf に比べて極端に大きくなってしまい、文節の削除候補を挙げる際に、大きな影響を与えていることが判明した。 idf の重みを抑えた値を用いたり、内容語のみの $tf \cdot idf$ を用いる等、 $tf \cdot idf$ の利用方法や、他の重要度を測る尺度について今後検討する必要がある。

4. まとめ

本稿では帰納的学習を用いた議事録自動生成手法を提案した。本手法は重要文抽出、文体変換、文内短縮を経て議事録を生成する。帰納的学習を用いることにより、人手で規則を与える場合に比べ、議事録作成者の習性に沿った学習を行うことが可能である。評価実験を行った結果、 $tf \cdot idf$ を要素とした単語頻度ベクトルにおいて、人間が作成した議事録とのコサイン距離が、ベースライン手法の0.3788という値に比べ、0.4108という比較的良好な結果を得た。また、文体変換、文内短縮を行うことにより、コサイン距離の結果は向上し、単語ベースでの比較における有効性が確認された。今後の予定としては、実験データを増やし、性能がどこまで向上するかの確認、読みやすさ等につ

いての主観評価による評価、及び、文内短縮での重要語の推定方法について検討する予定である。

参考文献

- [1] 奥村 学, 難波 英嗣, “テキスト自動要約に関する最近の話題”, 自然言語処理, Vol.9, No.4, pp.97-116 (2002)
- [2] G. Murray, S. Renals, J. Carletta, and J. Moore, “Evaluating automatic summaries of meeting recordings”, ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, pp.33-40 (2005)
- [3] 小林 竜己, “談話の局所・中位構造を利用したチャット会議ログからの議事録自動生成”, 人工知能学会研究会資料, SIG-SLUD-A203-05, pp.29-34 (2003)
- [4] 山本 和英, 安達 康昭, “国会会議録を対象とする話し言葉要約”, 自然言語処理, Vol.12, No.1, pp.51-78 (2005)
- [5] K. Araki and K. Tochinai, “Effectiveness of Natural Language Processing Method Using Inductive Learning”, IASTED International Conference ARTIFICIAL INTELLIGENCE AND SOFT COMPUTING, pp.295-300 (2001)
- [6] “茶筌”, <http://chasen.aist-nara.ac.jp/hiki/ChaSen/>
- [7] 阿部 里美, 荒木 健治, “汎用的要約システムの議事録に対する適応性能の評価”, 平成17年度電気・情報関係学会北海道支部連合大会講演論文集, 179 (2005)
- [8] 阿部 里美, 荒木 健治, “表現の特徴を用いた議事録自動生成システムの性能評価”, 言語処理学会第12回論文集, p.1079-1082 (2006)
- [9] “CaboCha”, <http://chasen.org/~taku/software/cabocho/>
- [10] “札幌市男女共同参画施策ホームページ”, <http://www.city.sapporo.jp/shimin/danjo/>
- [11] “NTCIR-TSC”, <http://ir-www.pi.titech.ac.jp/tsc/>