形態素解析とチャンキングの組み合わせによるフィラー/言い直し検出

浅原 正幸　　松本 裕治

奈良先端科学技術大学院大学 情報科学研究科

{masayu-a,matsu}@is.aist-nara.ac.jp

Abstract

We propose a novel filler/disfluency identification method for transcription of spontaneous speech in Japanese. Our method is based on Japanese morphological analysis and chunking. Firstly, input sentences are analyzed with redundant outputs by a statistical morphological analyzer. Since fillers and disfluencies produce ambiguity in morphological analysis, we do this so as to take into account several possible roles for each character in the input. Secondly, a support vector machine-based chunker detects some ambiguous points as fillers or disfluencies. Although it cannot detect disfluency of function words satisfactorily, it achieves high performance for fillers and disfluencies of content words.

1 はじめに

書き言葉と異なり，話し言葉の形態素解析は依然高い精度が得られていない．文献 [7] では，話し言葉の形態素解析における問題点として「フィラー」「非流暢性」「非文法性」「話し言葉に固有の表現」の四つをあげている．彼らは形態素解析器の話し言葉適応手法として，少量の話し言葉を書き言葉のデータに混ぜて利用する手法を提案している．この手法では，あげられた四つの問題のうち，厳密に対処されているのは「非文法性」のみである．その後「日本語話し言葉コーパス」[1] (以降「CSJ コーパス」) が整備されると同時に書き言葉にも話し言葉にも利用できるような形態素解析器用の辞書 UniDic [6] の開発が進められている．UniDic では，話し言葉に固有の表現にも対応できるよう，語彙，品詞，活用情報，発音情報の整備を行っている．統計モデルは CSJ コーパスから推定され「非文法性」「話し言葉に固有の表現」の両方に対処される形態素解析辞書として期待されている．

しかしながら,「フィラー」や「非流暢性」の問題に関して，上の先行研究では完全には対処できない．統計的形態素解析器はマルコフモデルに基づくものが多いが，フィラーや言い直し出現箇所で形態素認定規則として与えられる文脈が打ち切られ，解析誤りを誘発する原因になっている．また，あらかじめ形態素解析器に入力する前に，書き起こしの時点でフィラーや言い直しのタグを付与することが考えられるが，随時人手でこれを行うことは煩雑である．自動的にフィラーや言い直し出現箇所を検出する方法が求められている．

本稿では，この問題に対し，形態素解析器とチャンカーを用いてフィラーや言い直し出現箇所を検出する手法を提案する．提案手法では，まず入力テキストを形態素解析器で解析する．形態素解析器は入力テキストに対し，冗長解析結果を出力する．次に冗長解析結果を文字単位に分割し，各文字にその文字が属していた単語中の位置とその単語の品詞情報が付与される．最後にチャンカーを用い，その冗長解析結果を基に，形態素解析器にとって解析誤りとなりやすい文字列（つまり，フィラーや言い直し）をチャンクとして切り出す．この手法により，自動的に高精度のフィラーフィルターを構成することが可能になった．

2 節では検出対象となるフィラーと言い直しについて述べる．3 節では提案手法の詳細について述べる．4 節では「日本語話し言葉コーパス」を用いた評価実験を提示し，最終節でまとめと今後の課題について述べる．

2 フィラーと言い直し

本節では検出対象であるフィラーと言い直しについて述べる．

フィラーは発話中の空白を埋める発話である．典型的なフィラーとして長音符号がついた単母音や「あの」「それで」などの接続詞があげられる．フィラーはリストを作成し認定することが多く，CSJ コーパスでもフィラーとして認定されるもののリストを作成してタグづけを行っている．しかしながら，フィラーのリストを単純に形態素解析用辞書に追加するだけでは，精度良くフィラーを認定することができない．

言い直しは一度発音されその後に言い直される語や語断片である．あらかじめどの単語が言い直されるかが予測できないため，フィラーと異なりリストを作成することは不可能であり，形態素解析の枠組ではこれを検出することはできない．

CSJ コーパスでは，フィラーと言い直しなどの現

(F えーと) 音声と文字 (D2 の) との関係を..
(D さ) 三十秒間における

図 1: 「フィラー」および「言い直し」の例

象が談話書き起こしの上にタグづけされている．図 1 に例を示す．(D) は自立語の言い直し，(D2) は付属語の言い直し，(F) はフィラーを意味する．

3 提案手法

本節ではフィラー/言い直し出現箇所同定に対する提案手法について詳説する．本研究の目標は前節に示したフィラーと言い直し出現箇所を談話書き起こしテキストに対し自動タグづけを行うことである．提案手法は以下の三ステップによる．

1. 冗長的に入力テキストを形態素解析する
   ここで「冗長的に」とは，入力テキスト中の各文字に対してその文字を含む可能な形態素を複数出力することを意味する
2. 文字単位に分割し，字種情報，前後の文字との発音の類似度，各文字が属する形態素の情報と，その形態素中における文字の位置情報を付与する
3. 文字に付与された情報を手がかりに，形態素解析の弱点であるフィラー/ 言い直しをサポートベクトルマシンに基づくチャンカーを用いて検出する

まず，統計的形態素解析器を用い入力テキストを冗長的に解析する．フィラーおよび言い直しが出現する場合，形態素解析器はその出現箇所で解析を誤りやすく，より曖昧な冗長解析結果を出力する．この曖昧さを文字単位に素性として展開し，これをチャンカーの入力とする．チャンカーはこれらの素性を基に形態素解析器の弱点であるフィラーもしくは言い直しが出現したかどうかを同定する．以下，各ステップについて説明する．

3.1 冗長的な形態素解析

本手法で用いる日本語形態素解析はマルコフモデルに基づく．形態素解析は入力文 S の単語列 W に対する品詞タグ列 T を決定することと定義できる．目標は次の確率値を最大にするような品詞タグ列 T を発見することである．日本語や中国語の場合には，入力が文字列となるため，可能な単語列を全て展開した上で品詞列同定と単語列同定を同時に行うことになる．

$$T = arg\max_{T} P(T|W).$$

ベイズの定理を利用して，$P(W|T)$ は品詞タグ列の生起確率と単語列の生起確率として展開することができる．

$$arg\max_{T} P(T|W) = arg\max_{T} P(W|T)P(T).$$

単語生起確率はその品詞タグからのみに，品詞タグ生起確率は bi-gram モデルのみに制限して近似をする．これらの確率値はタグ付きコーパスの頻度から最尤推定される．推定されたパラメータを利用して，動的計画法の一種である Viterbi algorithm により，単語列 W に対して出現確率最大の品詞タグ列 T を決定する．実際の計算には確率の対数を取り，コスト（対数尤度）に変換して，可能な単語/品詞列からコスト和が最小になるようなものを選ぶことにより解析を行う．

本手法で用いる冗長解析は，最適解から設定したコスト幅のしきい値以内の n 次解を出力することによる．各文字位置において，その文字を含む文頭からのコスト和が小さい順に n 次解として形態素を出力する．尚，コスト和がしきい値を越えて異なる場合には，その解を出力しない．本手法ではしきい値として，確率モデルを推定する際，最低確率である事象に割り当てられるコストを用いた．

3.2 チャンカーの学習に用いる素性

表 1: 冗長形態素解析結果に付与するタグ

タグ	タグの説明
S	一文字で形態素を構成するもの
B	形態素（二文字以上）中の最初の文字
E	形態素（二文字以上）中の最後の文字
I	形態素（三文字以上）中の最初の文字でも最後の文字でもないもの

冗長的な形態素解析により認定された形態素を文字単位に分割する．各文字に対し，属していた形態素の品詞情報とその形態素中の位置の情報を素性として付与する．位置の情報には，表 1 に示すタグを用いる．属していた品詞の情報と位置の情報との二つ組を素性として導入する．さらに，字種の情報を素性として導入する．字種は「空白」「アラビア数字」「英字小文字」「英字大文字」「ひらがな」「カタカナ」「その他（漢字）」の七種類を導入する．

言い直しを検出するためには，発音の繰り返しに関する素性を導入する必要があるだろう．そこで，ひ

位置	文字	字種	発音の差				品詞（一次解）	品詞（二次解）	品詞（三次解）	フィラータグ
			-2	-1	+1	+2				
$i-2$	い	ひらがな	0	0	0	1	フィラー-S	形容詞-一般-S	動詞-一般-S	D-B
$i-1$	短	その他	0	0	0	0	形容詞-一般-B	形容詞-一般-B	接頭詞-S	O
i	い	ひらがな	1	0	3	0	形容詞-一般-E	形容詞-一般-E	フィラー-S	O
$i+1$	え	ひらがな	0	3	0	0	フィラー-B	*	*	
$i+2$	ー	カタカナ	0	0	0	0	フィラー-E	*	*	

図 2: 展開された素性

らがなに関しては固定長の文脈内に出現する文字間の「発音の差」に関する素性を入れる.「発音の差」は次の四つの値を持つ：1: 同じ発音，2: 母音を共有する，3: 子音を共有する，0: 上記三つ以外.

この発音の差に関する素性を当該文字と前後二文字以内の文字との間に定義する．図 2 に展開された素性の例を示す.

3.3 サポートベクトルマシンによるチャンキング

最後に展開された素性を基に，形態素解析器の弱点をチャンカーでフィラーもしくは言い直しとして抽出する．チャンキングではサポートベクトルマシン [3] を基にしたチャンカー yamcha [5] を利用した．以下にサポートベクトルマシンを用いたチャンキングについて述べる.

サポートベクトルマシンは，n 次元素性ベクトル $\mathbf{x}_t$ と正・負の二値ラベル y_t の二つ組 $(\mathbf{x}_t, y_t)$ で表現される訓練事例から，未知の $\mathbf{x}$ に対し正・負のラベルを正しく付与するような決定関数を与える二値分類器である．与えられる決定関数は次のように書くことができる：

$$f(\mathbf{x}) = sign(\sum_{\mathbf{z}_i \in SV} \alpha_i y_i K(\mathbf{x}, \mathbf{z}_i) + b.)$$

$f(\mathbf{x}) = +1$ は $\mathbf{x}$ がある特定のクラスであることを示し，$f(\mathbf{x}) = -1$ は $\mathbf{x}$ がそのクラスでないことを示す．ベクトル $\mathbf{z}_i$ は二値分類に必要な代表的な事例であり，サポートベクトルと呼ぶ．このサポートベクトルは二次計画法により決定される．$K(\mathbf{x}, \mathbf{z})$ はベクトルを高次元空間へと写像する関数であり，カーネル関数呼ばれる．本手法では二次の多項式関数 $K(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x} \cdot \mathbf{z})^2$ をカーネル関数として用いる．これにより，素性の二つまでの組み合わせを考慮した学習が可能になる.

サポートベクトルマシンは正例・負例を分類する二値分類器であり，チャンキング抽出規則を学習するために三つのクラスに分類する多値分類に拡張する必要がある．本手法では，k 個のクラスから任意の二つのクラスに関する二値分類器を ${}_kC_2$ 個構築する pairwise 法を用いた.

表 2: チャンキングに利用するタグ

タグ	タグの説明
B	チャンクのはじまり
I	チャンクの内側（B 以外）
O	チャンクの外側

整形した冗長形態素解析結果を文字単位にチャンキングする．チャンキングのために付与するチャンクタグは IOB2 タグ [2] と呼ばれ，表 2 に示す．CSJ コーパスで定義されているフィラータグとこのチャンクタグとの二つ組を各文字に付与することによりチャンキングを行う．チャンキングは 3.2 節で示した素性をサポートベクトルマシンに与え，その出力クラスを基に文頭もしくは文末から一方向に決定的に行われる．図 2 に前後二文字文脈にある文字，字種，発音の差および冗長形態素解析結果三次解までの品詞を用いた場合に利用される素性を示す．ここでタグ D-B は,「短い」によって言い直された発話を表している．この例では，位置 i におけるタグ O を推定するために，実線の内部にあるものを素性として利用する.

4 評価実験

実験には「日本語話し言葉コーパスモニター版 (2002)」を用いた．非言語的イベントのみの発話を取

表 3: 実験結果 (UniDic)

素性「発音の差」なし

	正方向（左 → 右）			逆方向（左 ← 右）		
タグ	再現率	精度	F 値	再現率	精度	F 値
(D)	59.9%	41.4%	48.9	69.1%	46.6%	52.5
(D2)	7.3%	32.3%	11.9	8.0%	35.5%	13.0
(F)	93.3%	93.8%	93.6	93.7%	93.1%	93.7
全て	87.3%	83.0%	85.1	87.7%	84.8%	86.3

素性「発音の差」あり

	正方向（左 → 右）			逆方向（左 ← 右）		
タグ	再現率	精度	F 値	再現率	精度	F 値
(D)	60.9%	40.4%	48.6	61.1%	44.7%	51.6
(D2)	10.2%	25.5%	14.4	10.2%	25.5%	14.6
(F)	93.0%	93.6%	93.3	92.6%	93.4%	93.0
全て	87.3%	82.2%	84.6	86.9%	83.6%	85.3

り除いた 95418 の発話を実験に用いた．80% (77058 発話，54 話者，(D):7102, (D2):433, (F):39504) を訓練データとして，20% (18360 発話，13 話者，(D):1633, (D2):137, (F):9299) をテストデータとして用いた．

形態素解析器として茶筌 [8] を用いた．形態素解析器の統計モデルは CSJ コーパスの品詞タグを UniDic 品詞体系 [6] に変換したもの（以降 UniDic と呼ぶ）から推定した．チャンカーとして yamcha [5] を用いた．オープンテストを実施するため，テストデータにしか出現しない語は UniDic の語彙から除いて実験を行った．発音の差を導入するものと導入しないものとの二種類の設定で実験を行った．また，チャンカーの解析方向に関し，正方向と逆方向の二種類の設定で実験を行った．実験結果として五分割交差検定による再現率，精度，F 値 ($\beta = 1$) を表 3 に示す．

チャンカーの解析方向に関して若干逆向きにチャンキングを行った方が精度が良い．フィラーのエントリを含む辞書に基づく形態素解析器のみによるフィラー検出結果は 91.5 (F 値) をベースラインの数値として用いることができるだろう．このベースラインの数値に対し，提案手法はフィラー検出結果 93.7 (F 値) と若干数値を上げるだけではなく，既存の手法では認識できなかった内容語の言い直し現象も 52.5 (F 値) の精度で抽出することが可能である．しかしながら，表層形が短く，頻度が少ない機能語の言い直し現象は良い精度で検出することができなかった．「発音の差」の素性は導入すると機能語の言い直し現象の抽出精度を若干向上させることができたが，全体の精度を下げる結果となった．

同様の実験を茶筌とともに配布されている ipadic [4] を用いて行った．結果を表 4 に示す．UniDic が登録語 18,000 語に対して，ipadic が登録語 240,000 であるため，ipadic を用いた方が未知語が出現しにくい．結果，ipadic の方が良い精度を得ることができた．このことから，現在のところ形態素解析器の弱点でかつフィラー以外の箇所を言い直しの出現箇所として推定していると考えるのが妥当であろう．言い直しと真の未知語とを区別するような素性が必要であると考える．

表 4: 実験結果 (ipadic)

素性「発音の差」なし						
	正方向（左 → 右）			逆方向（左 ← 右）		
タグ	再現率	精度	F 値	再現率	精度	F 値
(D)	55.9%	63.9%	59.6	54.6%	64.3%	59.1
(D2)	5.8%	33.3%	9.9	7.3%	34.5%	12.0
(F)	92.8%	93.3%	93.0	92.9%	93.0%	93.0
全て	86.3%	89.3%	87.7	86.2%	89.2%	87.7
素性「発音の差」あり						
	正方向（左 → 右）			逆方向（左 ← 右）		
タグ	再現率	精度	F 値	再現率	精度	F 値
(D)	56.0%	62.7%	59.1	56.5%	63.0%	59.6
(D2)	11.7%	31.4%	17.0	13.1%	35.3%	19.1
(F)	92.8%	93.6%	93.2	93.0%	93.5%	93.2
全て	86.4%	89.1%	87.7	86.7%	89.0%	87.8

5 おわりに

形態素解析器の弱点をチャンカーで調査することにより，フィラーおよび言い直しの出現を検出する手法を提案した．本手法では依然として言い直し（特に機能語）を高精度で検出できているとは言い難い．今回の実験では，漢字の読みの展開を行っていない．今後，読みの展開を行い，言い直しに対する素性を豊かにすることにより，さらなる精度向上を目指したい．

謝辞

「日本語話し言葉コーパス」の整備に携わっている方々および UniDic 体系の品詞・活用形・活用型定義を提供していただいた千葉大の伝康晴氏に感謝の意を表します．

参考文献

[1] K. Maekawa, K. Hanae, S. Furui, and H. Isahara. Spontaneous Speech Corpus of Japanese. In *LREC2000*, pp. 947–952, 2000.

[2] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-bases learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pp. 83–94, 1995.

[3] V.N. Vapnik. *Statistical Learning Theory.* A Wiley-Interscience Publication, 1998.

[4] 浅原正幸, 松本裕治. IPADIC ユーザーズマニュアル. 奈良先端科学技術大学院大学, 2002.

[5] 工藤拓, 松本裕治. Support Vector Machine を用いた Chunk 同定. 自然言語処理, Vol. 9, No. 5, pp. 3–23, 2002.

[6] 伝康晴, 宇津呂武仁, 山田篤, 浅原正幸, 松本裕治. 話し言葉研究に適した電子化辞書の設計. 第 2 回「話し言葉の科学と工学」ワークショップ講演予稿集, pp. 39–46, 2002.

[7] 松本裕治, 伝康晴. 話し言葉の形態素解析. 情報処理学会研究会報告 (自然言語処理研究会), No. 2001-NL-143, pp. 49–54, 2001.

[8] 松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸. 形態素解析システム『茶筌』version 2.2.9 使用説明書. Technical report, 奈良先端科学技術大学院大学, 2002.