視覚障害者用音声ブラウザのためのウェブページ解析

加藤　邦彦　　白井　清昭

北陸先端科学技術大学院大学 情報科学研究科

{k-katou, kshirai}@jaist.ac.jp

1 はじめに

視覚障害者がWWWを閲覧するためのソフトとしてスクリーンリーダーや音声ブラウザなどがある．これらはウェブページの情報を音声によって伝えることによって視覚障害者のウェブページ閲覧を補助するツールである．テキストを音声合成によって読み上げる機能を中心とし，キー操作によるページ内の移動やリンク先ページへの移動も可能である．既に実用化され，製品として販売されているツールもある．

既存の多くの視覚障害用音声ブラウザの主な問題点は，常にウェブページの先頭から順にテキストを読み上げるという点である．そのため，ユーザの知りたい情報が読み上げられるまでに，広告などの興味のない情報を延々と聞かなければならないことが多々ある．目が見える人なら，レイアウトや文字色などの視覚的情報から知りたい情報が書いてある場所を素早く見つけることも可能である．しかし，音声ブラウザの場合はレイアウト等の情報は失われるため，ウェブページのどこに知りたい情報があるかを推測することは困難である．

本研究では，上記の問題を解決するため，ウェブページの構造解析を基盤とした視覚障害者用音声ブラウザのナビゲーション機能を実現することを目的とする [2].

2 提案システム

本節では，本研究で提案する視覚障害者用音声ブラウザのための2つのナビゲーション機能について説明する．

2.1 ウェブページの構造解析に基づく不要テキスト読み飛ばし

ウェブページの構造を解析し，ページをいくつかのセグメントに分割する．ここでセグメントとは，ウェブページにおけるテキストの構造上あるいはレイアウト上のまとまりを指す．ウェブページとその内部にあるセグメントの例を図1に示す．図1中の枠線がひとつのセグメントを表わす．

音声ブラウザがウェブページ上のテキストを読み上げ，セグメントの開始位置に到達したとき，セグメントの存在とその内容をユーザに知らせ，セグメント内のテキストの読み上げを続行するかどうかを選択させる．ユーザは，そのセグメントの内容に興味がなければそのセグメントを読み飛ばせる．例えば，図1のページにおいて，グルメ情報を知りたいユーザは広告や「買う」「知る」といったセグメントの読み上げを省略できる．

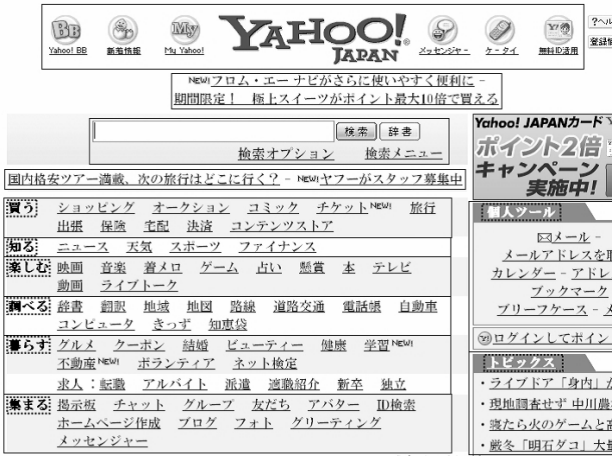

図 1: ウェブページのセグメントの例

ここで問題となるのは，セグメントをどのように検出するかということと，セグメントの内容をどのようにユーザに伝えるかという点である．本論文では，前者に焦点を当てている．その詳細については3節で述べる．後者については本論文の対象外とし，今後の課題とする．ただし，現在のところ以下のような手法を検討している．まず，セグメントの内容を伝える簡単な手法として，セグメントの先頭にある見出しの提示を考えている．例えば，図1の例では，セグメントの先頭に位置する「買う」や「知る」はそのセグメントの見出しと考えられる．ただし，セグメントの先頭が必ずしも見出しになっていない場合もある．このような場合は，既存の要約技術を応用し，セグメント内のテキストの中から代表的なキーワードを抽出して提示することが必要となる．

2.2 リンク先ページの読み上げ開始位置の推定

ユーザがリンクを辿るとき，リンク先ページの情報に関心があるとみなすことができるが，その情報はページの先頭に書いてあるとは限らない．そこで，リンクのアンカー文字列の情報を手がかりに，リンクが参照するリンク先ページ内位置を自動的に特定し，その位置から読

み上げを開始する．4 節で述べるように，リンクの参照位置は複数検出されるが，そのときは一番最初に出現する参照位置を読み上げ開始位置とする．また，読み上げ開始位置が適切でない場合は，キーボードによる操作などによりユーザは次の参照位置にスキップできる．これにより，リンクの参照位置より前にある不必要なテキストの読み上げを省略できる．

3 セグメント検出

本節では，2.1 項で述べたナビゲーション機能のためのウェブページのセグメント検出手法について述べる．まず，本研究ではウェブページのセグメントを以下のように定義する．

- 何らかの共通の属性を持つテキストの集まりである
- テキスト要素の数は 2 つ以上
  ここでテキスト要素とは HTML タグの間に狭まれた文字列を指す．あまりに小さいセグメントは読み飛しても有益ではないため，検出しない．
- セグメントは入れ子になってもよい
- ページ全体をカバーしなくてもよい
  従来のウェブページ構造解析手法の多くはページ全体の構造を把握することを目的としているが，本研究では不要箇所の読み飛しが目的なので，全てのテキスト要素が何らかのセグメントに属す必要はない．(セグメントに属さないテキスト要素はそのまま読み上げる)

セグメント検出までの処理の流れを以下に示す．

3.1 DOM に基づくセグメント検出

前処理として，Tidy [1]を用いて HTML 文書の整合性のチェック及び補正を行う．次に，ウェブページの DOM (Document Object Model) ツリーを得，表 1 の HTML タグを手がかりとしてセグメントの検出を行う．表 1 において，T_r はそのタグで囲まれている領域をセグメントとみなすタグである．一方，T_b はそのタグをセグメントの開始位置とみなすタグである．

セグメント検出アルゴリズムを以下に示す．DOM ツリー上の全てのノードを深さ優先でたどる走査を 2 回行う．1 回目の走査では，T_b に該当するタグのノードには boundary というマークをつける．また，T_r に該当するタグのノードについては，もしそのノードの下位に T_r に属すタグが存在しなければ range というマークをつける．これは，例えば table タグが入れ子で使われていた場合，一番下位にある table タグのみをセグメント境界とみなすことを意味する．

表 1: セグメント検出の手がかりとなるタグ

T_r	table, ol, dl, ul, p
T_b	h1, h2, h3, h4, h5, h6, hr

[1] http://tidy.sourceforge.net/

2 回目の走査では，検出したセグメントの開始位置と終了位置をコメントとして出力する．range にマークをつけたノードでは，その開始タグと終了タグの位置にセグメントの開始位置と終了位置を同時に出力する．bounrdary マークをつけたノードでは，そのノードの直前の位置にセグメントの開始位置のみを出力する．その後，boundary または range マークのついたノードを見つけたら，その直前の位置に先ほど開始位置を出力したセグメントの終了位置を出力する．

3.2 セグメント分割

DOM に基づく手法で検出されたセグメントが大きすぎる場合はセグメントの分割を行う．

イメージを手がかりとしたセグメント分割

自動検出されたセグメント内に同じイメージ (画像) が頻出するとき，そのイメージがセグメント分割の手がかりになる場合がある．例えば，図 2 において，枠線は自動検出されたセグメントを表わすが，その内部のレイアウトは箇条書きに近く，個々の箇条書き (「オリジナル・ベストセレクト」「大人のレストランガイド」など) をひとつのセグメントとみなすべきである．この場合，箇条書きのアイテムを表わす丸いアイコンがセグメントの境界を示唆していると考えられる．

まず，同一セグメント内で最も頻繁に現われるイメージを検出する．イメージの同一性の判定は img タグの src 属性によって行う．もし最頻出イメージの出現頻度がある閾値 (本研究では 4) 未満ならセグメントの分割は行わない．次に，最頻出イメージを境界とした仮のセグメントを検出する．検出された仮セグメントのテキスト要素の数が 2 以上なら，それを新たなセグメントとみなす．新たなセグメントが 1 つ以上検出されたとき，分割前の元のセグメントは除去する．

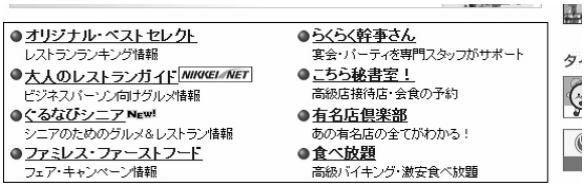

図 2: イメージを手がかりとしたセグメント分割

DOM の部分木を手がかりとしたセグメント分割

ここでは table タグを手がかりに検出されたセグメントをさらに分割することを考える．table タグがレイアウト目的で使用されている場合，table 全体が常にひとつのセグメントになるわけではない．table の各行，すなわち tr タグが囲む範囲や，各セル，すなわち td タグが囲む範囲がウェブページ上のひとつのまとまりとみなせる場合もある．ここでは，tr タグや td タグが囲む範囲の類似度を求め，類似度が高い場合，すなわち似ている領域がテーブルの行またはセル毎に現われている場合は，それらを新たなセグメントとして分割する．

まず，3.1 項の手法で table タグによって検出されたセグメントに対して，td タグを根とする DOM の部分木 ($s_1 \sim s_n$) を得る．次に，n 個の部分木の最大の共通部分木 (s_c) を求める．最後に s_1〜s_n の類似度を式 (1) のように定義する

$$Sim(s_1, \cdots, s_n) = \frac{size(s_c)}{\min_i size(s_i)} \quad (1)$$

ここで $size(s)$ は部分木 s の大きさであり，s に含まれるノード数と定義する．$Sim(s_1, \cdots, s_n)$ がある一定の閾値を越えれば，個々の td タグが囲む範囲を新たなセグメントとして検出する．分割前の元のセグメントは除去する．本研究ではこの閾値を 0.5 とした．また，td タグを根とする DOM の部分木集合に対してセグメント分割に失敗した場合は，tr タグを根とする DOM の部分木集合に対して同様の操作による分割を試みる．

3.3 セグメントのマージ

DOM に基づく手法 (3.1 項) で検出されたセグメントが小さすぎる場合がある．特に同一セグメントのヘッダと見られる部分が同じセグメントとして検出されない誤りが多く見られた．例えば，図 3 では「ニュース速報」とそれ以降のニュースの一覧が別々のセグメントとして検出されているが，これらはひとつのセグメントとして検出されるべきである．特に，2 節で述べたように，本研究では見出しの情報をセグメントの内容を表わす語としてナビゲーションに利用する予定である．したがって図 3 の「ニュース速報」のようなセグメントの見出しを表わすテキストに対してセグメントを正確に検出することは重要である．

セグメントのマージ手法を以下に述べる．まず，以下に挙げるタグで囲まれたテキスト要素はセグメントの見出しになりやすいと仮定し，ウェブページから検出する．

h1〜h6 のヘッドタグ, b, thead

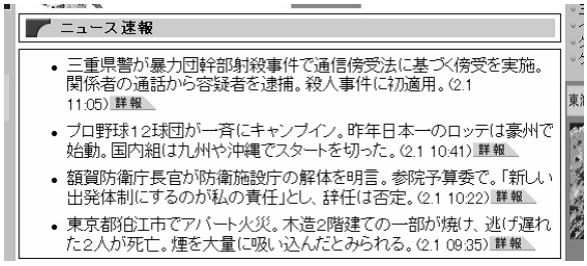

図 3: セグメントのマージ

このとき，テキスト要素がセグメントであるかどうかは区別しない．DOM に基づくセグメント検出 (3.1 項) でセグメントとして検出されなかったテキスト要素の中にも，後続のセグメントの見出しとみなせるものがあるからである．次に，検出したテキスト要素の直後にセグメントが存在すれば，そのテキスト要素とセグメントをマージして一つのセグメントとする．また，マージ前のセグメントは除去する．

4 リンクの参照位置の検出

本節では，2.2 項で述べたナビゲーション機能の実現のために，リンク先ページの中からリンクが指し示す位置を正確に検出する手法について説明する．

まずリンクのアンカー文字列を anc とする．次に，そのリンク先ページの全てのテキスト要素 t_i について，以下のいずれかの条件を満たすかどうかをチェックする．

- anc 中の部分文字列として t_i が存在する
- t_i 中の部分文字列として anc が存在する

このチェックは Perl のパタンマッチによって簡単に実現できる．この条件を満たす全てのテキスト要素 t_i をリンクの参照位置として出力する．ただし，t_i が短い文字列のときは誤判定を起こす可能性があるので，長さ 5 未満の t_i についてはパタンマッチを行わず，リンクの参照位置として検出しない．

5 予備実験

5.1 セグメント検出

3 節で述べたセグメント検出手法の評価を行った．まず，ウェブページの集合を 2 つ用意した．1 つは手法の検討に用いたウェブページの集合，もうひとつはそれ以外のウェブページの集合である．前者に対する評価はクローズドテスト，後者はオープンテストとみなせる．ページ数はそれぞれ 20 ページである．これらのウェブページに対して人手で正解となるセグメントを付与し，提案手法によって自動検出されたセグメントと比較した．結果を表 2 に示す．表 2 において，“提案手法” は 3 節

表 2: セグメント検出の実験結果

(クローズド)	R	P	PB	C
DOM only	0.281	0.333	0.465	0.55%
提案手法	0.442	0.442	0.644	1.43%

(オープン)	R	P	PB	C
DOM only	0.257	0.283	0.433	1.22%
提案手法	0.338	0.299	0.580	3.35%

で述べた手法を，“DOM only” は 3.1 項の手法のみを用いたときを表わす．また，“R” は再現率を，“P” は精度 (適合率) を，“PB” はセグメントの開始位置が正しければ正解とみなしたときの再現率を，“C” は正解のセグメントと領域が交差するセグメントの割合を表わす．PB は，セグメントの開始位置さえ正しければ，終了位置が合っていなくても不要なテキストを読み飛ばすという目的を考えれば一定の成果を挙げていると考え，導入した評価尺度である．

表 2 より，DOM だけを用いる手法と比べて，セグメントの分割やマージによって正解率が向上したことがわかる．また，正解率は実用的な音声ブラウザとして実装できるほど十分に高いとは言えず，更なる改良が求められる．とはいえ，正解のセグメントと交差するセグメントの割合 (C) はごく小さいことから，セグメント検出手法自体にはそれほど大きな問題はないと考えられる．

5.2 リンクの参照位置の検出

4 節で述べたリンクの参照位置の検出手法の評価を行った．100 個のリンクとリンク先ページの組を用意し，リンクが参照している位置をリンク先ページの中に人手で付与し，正解データとした．実験の結果，再現率は 67%，適合率は 18%となった．また，ページの先頭から読み上げを開始する場合にはリンクの参照位置に到達するまでに 75.1 個のテキスト要素を読み上げるのに対し，本手法で検出した位置を順番にたどる場合に読み上げが必要なテキスト要素の数は 2.2 個であり，不要なテキストの読み上げを大幅に省略できることが確認された．

現在のリンクの参照位置の検出手法は非常に単純である．正解率そのものの評価よりも，単純な文字列のマッチングによる正解率の目安を示したことが重要と考える．検出に失敗した事例の多くは，予想通り，リンクのアンカー文字列と同じ内容を表わすテキストが別の表現で表わされている場合である．したがって，単語ベクトルによってテキスト間の類似度を計算したり，何らかの言い換え技術を導入する必要がある．

6 関連研究

視覚障害者のウェブページ閲覧の支援を目的とした研究は既にいくつか行われている [1, 3]．しかし，本研究のようにテキストの読み上げをスキップしたり，読み上げ開始位置を変えることによる不要なテキストの読み飛ばしに焦点を当てた研究はまだない．

ウェブページの構造解析については数多くの研究が成されている．基本的には Yu ら [6] のように DOM の情報や HTML タグを手がかりとする手法が多いが，南野ら [5] のようにページ内に現われる要素系列の繰り返しを手がかりとする手法や，松本ら [4] のようにページの階層構造を日本語の係り受け構造に見立てて解析する手法などもある．本研究は，これらの手法の利点を組み合わせたウェブページ構造解析手法の実現を目指している．本論文でも，3.1 項の手法は DOM ツリーに基づく手法，3.2 項の手法は繰り返し構造の検出に基づく手法にヒントを得ており，両者の統合を目指した結果である．

7 おわりに

今後の課題として，セグメント検出手法やリンクの参照位置の検出手法の正解率の向上が挙げられる．特にセグメント検出については，3 節のセグメントの定義ではセグメントは入れ子になってもよいとしているが，現在の手法は入れ子状のセグメントを出力しない．したがって，手法の再帰的適用などによりセグメントの入れ子構造を正確に検出する必要がある．最終的に 2 節で述べたナビゲーション機能を実装した音声ブラウザを実装することも残された重要な課題である．

参考文献

[1] 浅川智恵子, 伊藤隆. 視覚障害者向け Web アクセスシステムにおける html タグの音声変換方式について. 自然言語処理シンポジウム, 1997.

[2] 加藤邦彦. 視覚障害者用音声ブラウザのためのウェブページ構造解析. Master's thesis, 北陸先端科学技術大学院大学, 3 2006.

[3] 前田潤治, 高木啓伸, 福田, 健太郎, 浅川智恵子. アノテーションに基づくウェブページのダイジェスト手法. 電子情報通信学会技術研究報告 (福祉情報工学) WIT2001-14, pp. 25–30, 1999.

[4] 松本吉司, 乾健太郎, 松本裕治. Web ページのテキストセグメント階層構造の抽出. 言語処理学会第 11 回年次大会論文集 (B1-4), pp. 49–52, 2005.

[5] 南野朋之, 齋藤豪, 奥村学. 繰り返し構造を用いた Web ページの構造化に関する研究. 自然言語処理研究会 2003-NL-154, 2003.

[6] Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the International WWW Conference*, 2003.