

読みやすさと視覚的バランスを両立した
改行・改頁位置の推定

筑波大学
図書館情報メディア研究科
2022年03月
大峠 和基

読みやすさと視覚的バランスを両立した
改行・改頁位置の推定

Detection of break positions for text
considering both readability and visual balance

学籍番号：201921624

氏名：大峠 和基

Otao Kazuki

雑誌・広告・動画などのビジュアルデザインにおいてテキスト要素を配置する際、表示できる領域が限られるためにテキスト要素に対して適切な位置に改行・改頁を挿入する必要がある。この際、読者に負担なくテキスト要素を読ませること、デザイン全体のバランスを崩さないことの2点を満たす必要があり、改行・改頁を挿入する工程は専門的なスキルが必要とされてきた。

本稿では文字列を入力としたとき、上述した2点の条件のもとで改行・改頁位置を推定する手法を提案する。具体的には、文字数や行数を制約として与え、可読性と視覚的バランスを考慮した評価関数を最小化する組み合わせ最適化問題として定式化する。従来よりテキスト要素の配置や大きさの最適化、可読性のみに焦点を当てた改行位置推定に関する研究がなされてきたが、自動デザインにおける視覚的バランスを考慮した改行位置推定の手法は知られていない。

提案手法及びベースラインとなる簡便な手法を実装し、クラウドソーシングを用いた利用者実験を行い、提案手法による改行・改頁の付与が可読性と視覚的バランスを向上させ、文章の読解速度や理解度の改善に寄与することを明らかにした。提案手法による応用の一例として、自動字幕システムとデザイン支援システムをスマートフォン上とWeb上にそれぞれ実装し、動的なデザイン生成や制作時間の短縮を可能にすることを確認した。

以上述べたように、日本語文章の可読性と視覚的なバランスを考慮して改行・改頁位置を推定する提案手法は、ビジュアルデザインにおける制作支援にとどまらず、個々の利用者や利用環境に適応した個人化リーフレットの動的な生成など、制作工程に人手介入を必要としない **End to End** なビジュアルデザインの自動生成に寄与することが期待される。

主研究指導教員：佐藤 哲司

副研究指導教員：藤澤 誠

目次

第1章	序章	1
第2章	関連研究	6
2.1	ビジュアルデザインにおけるレイアウト推定	6
2.2	可読性向上のための読点および改行位置推定	9
2.3	E2Eなビジュアルデザインの自動生成に向けた周辺研究	10
2.4	本稿の立ち位置	10
第3章	提案手法	12
3.1	ビジュアルデザインのための改行・改頁	12
3.2	改行・改頁位置推定	15
3.2.1	改行可能性推定	16
3.2.2	評価関数	18
3.2.3	最適化	19
3.3	結果	22
第4章	評価実験	24
4.1	短文における視覚的バランスの検証	24
4.1.1	デザイン	24
4.1.2	結果	27
4.1.3	考察	28
4.2	長文における可読性と理解度の検証	29
4.2.1	デザイン	29
4.2.2	結果	34
4.2.3	考察	36
第5章	応用	38
5.1	自動字幕システム	38
5.1.1	背景	38
5.1.2	システム構成	39
5.1.3	結果	39
5.2	デザイン支援システム	40
5.2.1	背景	40

5.2.2	システム構成	40
5.2.3	結果	40
第 6 章	議論と今後の展望	42
6.1	境界に対する改行可能性の推定精度	42
6.2	探索のオーダー	42
6.3	ビジュアルデザインのためのデータドリブンな改行位置推定	43
6.4	多言語への適用	43
第 7 章	結論	44
	謝辞	46
	参考文献	47

目 次

1.1	ビジュアルデザインのコンテンツ例.	2
1.2	ビジュアルデザインを構成するグラフィック要素とテキスト要素の例.	4
2.1	顕著性マップの生成.	7
2.2	Yang らによるテキスト要素の自動配置.	8
2.3	Donovan らによるユーザの編集に対してインタラクティブにレイアウトを提示するシステム.	9
3.1	改行位置推定の基本的な考え方.	12
3.2	文字列を塊で表した例.	13
3.3	異なる改行パターンをグラフィクス要素の上にオーバーレイして表示したときの様子.	14
3.4	行数 2 行の制約のもとで考えられる改行パターン.	15
3.5	改行可能性 p の推定結果.	18
3.6	提案手法を 4 つの文章に適用した結果.	23
4.1	短文における視覚的バランスの検証用に構築した Web サービス.	26
4.2	6 段階評価をバイナリ化した結果.	27
4.3	長文における可読性と理解度の検証のために構築した Web サービス.	30
4.4	テキストウィンドウの操作.	31
4.5	チュートリアル画面.	33
4.6	読解時間と正答数の関係.	34
5.1	自動字幕システムの全体像.	39
5.2	制約を変化させたときの自動字幕生成の様子.	40
5.3	制約を変化させたときの自動改行の様子.	41

第1章 序章

ビジュアルデザインとは、イラストや画像，CGなどの視覚的な表現を使って顧客や大衆にメッセージを伝達することを目的としたクリエイティブの総称であり¹，図 1.1 に示すように雑誌・広告・ポスターなどが代表的な例である．その定義に基づけば新商品のプロモーション動画やモーショングラフィクスのようなアニメーション，企業の Web サイト，対外発表や内部資料として用いられるプレゼンテーション用のスライドなどもビジュアルデザインに含まれる．特にデジタルデバイスの普及した現代においては，ビジュアルデザイン分野におけるクリエイティブの重要性は日々増加しておりその影響力は計り知れない．

¹<https://ja.wikipedia.org/wiki/ビジュアルデザイン> (最終閲覧: 2021 年 12 月)



図 1.1: ビジュアルデザインのコンテンツ例。(a) 雑誌。(b) ポスター。(c) バナー広告。(d) プレゼンテーション用スライド。(e) 動画の字幕。(コンテンツ例は canva.com および unsplash.com から引用。)

ビジュアルデザインのコンテンツ制作では、質の高いクリエイティブを短い期間で納品しなければならないケースが多い。例えば、動画投稿サイト YouTube² には1日あたり約10億時間もの動画が毎日投稿されており³、視聴数を増やすためにはできるだけ短い間隔で高いクオリティの動画を投稿することが要求される。デジタル広告ではWebサイトやモバイルアプリ上の広告表示サイズの違いにより、デザイナーは同じ商品を訴求するために縦横率の異なる複数のクリエイティブを用意する必要がある。クリエイターやデザイナーに限らずより一般的な例としては、企画発表や情報共有のために制作するプレゼンテーション用のスライドが上げられる。この場合、最初からスライドのデザインを作り込むのではなく、たたき台となるモックスライドを制作した後、議論を重ねながら完成に近づけていく工程を踏むためスライドを素早く制作するためのノウハウが必要である。

そのような背景から、ビジュアルデザインの制作工程の自動化が課題となっている。実際の工程は、手法やツールによって違いはあるものの、下書き・素材の配置・カメラやライティングの設定・エフェクト効果の設定・レンダリングといった多くの項目から成り立っているため、それらの一部分ずつを細分化して問題提起し、自動化を検討できる。Adobe Inc.⁴ の Adobe Photoshop⁵ や Adobe AfterEffects⁶ はデザイン制作工程をコンピュータサイエンスの要素技術によって支援するソフトウェアとして一般的にも普及している。これらのソフトウェアでは、実際には複雑な画像処理を行っているにも関わらず、ユーザはGUIを通じて人物のくり抜き・色彩調整・オブジェクト合成などの処理を簡単に行える。デザイン制作工程を細分化し自動化することは、SIGGRAPH を初めとするCG分野で過去数十年に渡って研究が続いており、商用・非商用のソフトウェアとして一般利用な形で提供され社会実装が進んでいる。

加えて、現代ではコンテンツを閲覧する個人にパーソナライズされたクリエイティブを動的に生成するケースも見受けられる。例えば、Web上でファッションについてのバナー広告を表示する際、その個人の性別や閲覧履歴などの個人情報を元に最適なジャンルの服飾画像を用意し、服飾画像に沿ったデザインの広告を自動生成する。動的に生成された広告は、個人の趣味趣向を反映しているため、視聴時間やクリック数、購入数などの数値の改善が期待される。このように、ビジュアルデザインにおける制作工程の自動化は、もはや単なる制作時間の短縮や初学者支援にとどまらず、デザインに対して人間の介入しない End to End (E2E) な自動生成の領域まで踏み込んでいる。コンテンツを閲覧する個人に合わせて最適なデザインを自動生成する技術は、今後も学術・商業の双方の分野で高い関心を集めることが想定される。

ところで、図1.2で示すように、ビジュアルデザインは大まかに主題となる被写体を示すグラフィクス要素と、キャッチコピーや訴求文を示すテキスト要素から成り立っていると捉えられる。従来より、この分野ではグラフィクス要素に対してテキスト要素の配置関係や大

²<https://www.youtube.com/> (最終閲覧: 2021年12月)

³<https://www.youtube.com/trends/articles/report-sources-tr20/> (最終閲覧: 2021年12月)

⁴<https://www.adobe.com/> (最終閲覧: 2021年12月)

⁵<https://www.adobe.com/products/photoshop.html> (最終閲覧: 2021年12月)

⁶<https://www.adobe.com/products/aftereffects.html> (最終閲覧: 2021年12月)

きさ、色を最適化する研究がなされてきた [1]。ここで単なる書類と異なり、読者に負担なく文書を読ませる必要のあるビジュアルデザインでは、配置するテキスト要素に対して適切な位置に改行・改頁を加えて整形することで、読みやすさや視覚的印象を向上させることが重要である。しかしながら、テキスト要素の文章自体に対して、改行・改頁位置を最適化することに着目した研究は未だ例を見ない。テキスト要素を人手で配置する際は、文節や句読点の後に改行を挿入することや語句の途中では改行を挟まないことなど、ある程度の弱い規則が存在することが見受けられるが、明確で一意的な規則は存在しない。一方で、音声認識結果をスクリーン上にリアルタイムに表示する講演システムを想定して、アノテーションを付与したデータセットを用いて改行位置を学習する機械学習アプローチが提案されてきた [2]。しかしながら、自動で改行位置を推定する従来研究では、改行による可読性のみが焦点が当てられており、改行が及ぼす視覚的印象については議論されてこなかった。ビジュアルデザインでは、図 1.1 に挙げた例のように、テキスト要素を表示できる領域が限られており、その制限はデザインによって異なる。また、テキスト要素そのものがデザインを構成する一要素であるため、単に可読性を担保するだけでなく、視覚的なバランス感を保つことが重要であり従来の改行位置推定の手法とは考慮すべき項目が異なる。



図 1.2: ビジュアルデザインを構成するグラフィック要素とテキスト要素の例。

これらの理由から、可読性と視覚的バランスの双方を考慮したビジュアルデザインのためのテキスト要素への改行・改頁の自動挿入手法が求められる。本稿では、限られた領域内に文章を表示する際、読みやすく、テキスト要素全体のバランス感が保たれるという条件のもとで、改行・改頁位置を推定する。具体的には、可読性と視覚的バランスを考慮した評価関数を設計し、組み合わせ最適化により最も評価関数を最小化する改行・改頁パターンを選択する。その際、文字数・行数の制約を与えることができ、これはシステムやユーザが自由に変更することができる。

従来手法に対する提案手法の利点は以下の 3 点である。

1. 可読性と視覚的バランスを両立した改行・改頁位置の推定.
2. 行あたりの文字数や行数の制約をシステムやユーザが自由に変更可能.
3. 省メモリでリアルタイムに動作.

提案手法は、雑誌・ポスター・広告・スライド・アニメーションなどのビジュアルデザインに応用できるほか、上記の利点を生かして次のような事例に応用可能である。

- 音声認識による自動字幕への自動改行.
- Web サイトのテキスト要素の自動改行.
- 小型 LCD 上に表示するテキストの自動改行.
- メッセージアプリやゲームのテキストウィンドウへの自動改行.

本稿の構成は以下の通りである。2章では、ビジュアルデザインにおけるレイアウトの自動推定と、テキスト要素への句読点・改行の自動推定の2つの観点から関連研究を俯瞰する。3章では、ビジュアルデザインで目指すべき改行を例を踏まえながら議論し、議論の結果から得られた理想的な改行・改頁を自動で付与するための組み合わせ最適化手法について述べる。4章では、クラウドソーシングによる実験を通じて、提案手法による改行・改頁付与の結果が可読性のみならず視覚的印象を向上させることを検証する。5章では、提案手法によるアプリケーションの一例として、自動字幕システムおよびデザイン支援システムへ組み込む。最後に6章で提案手法の制限と今後の展望について議論する。

提案手法により、従来人手で行われていたテキスト要素への改行・改頁挿入を自動化し、デザイン制作工程の効率化を目指すとともに、E2Eなビジュアルデザインの自動生成における一要素技術の確立を目指す。

第2章 関連研究

2.1 ビジュアルデザインにおけるレイアウト推定

雑誌やポスターなどを主とするビジュアルデザインの分野では、テキスト要素とグラフィクス要素を含むレイアウトを自動で生成する手法が探求されてきた [3]。テキスト要素を挿入する際は、(1) 主題となるグラフィクス要素に被らない位置、(2) グラフィクス要素とテキスト要素の配置関係がデザイン全体としてバランスの取れる位置、の2点を満たす配置を探す必要がある。これらを満たすために顕著性マップ [4] がしばしば用いられる。顕著性マップとはグラフィクス要素のうち人間の目を通じて見た際に注目を集める領域を高い値、注目に値しない領域を低い値で表した2次元の1チャンネル画像である。Houらによる顕著性マップの実装は、OpenCVのcontribモジュールに導入されているため簡単に試すことができる。図2.1(a)に入力画像、図2.1(b)にHouらによる顕著性マップ生成の結果を示す。この入力画像は奥のティーポットやカップがボケがかかっており、手前のトーストに焦点があたっている。生成された顕著性マップではそれを反映するように手前のトーストの領域に高い値が集中している。

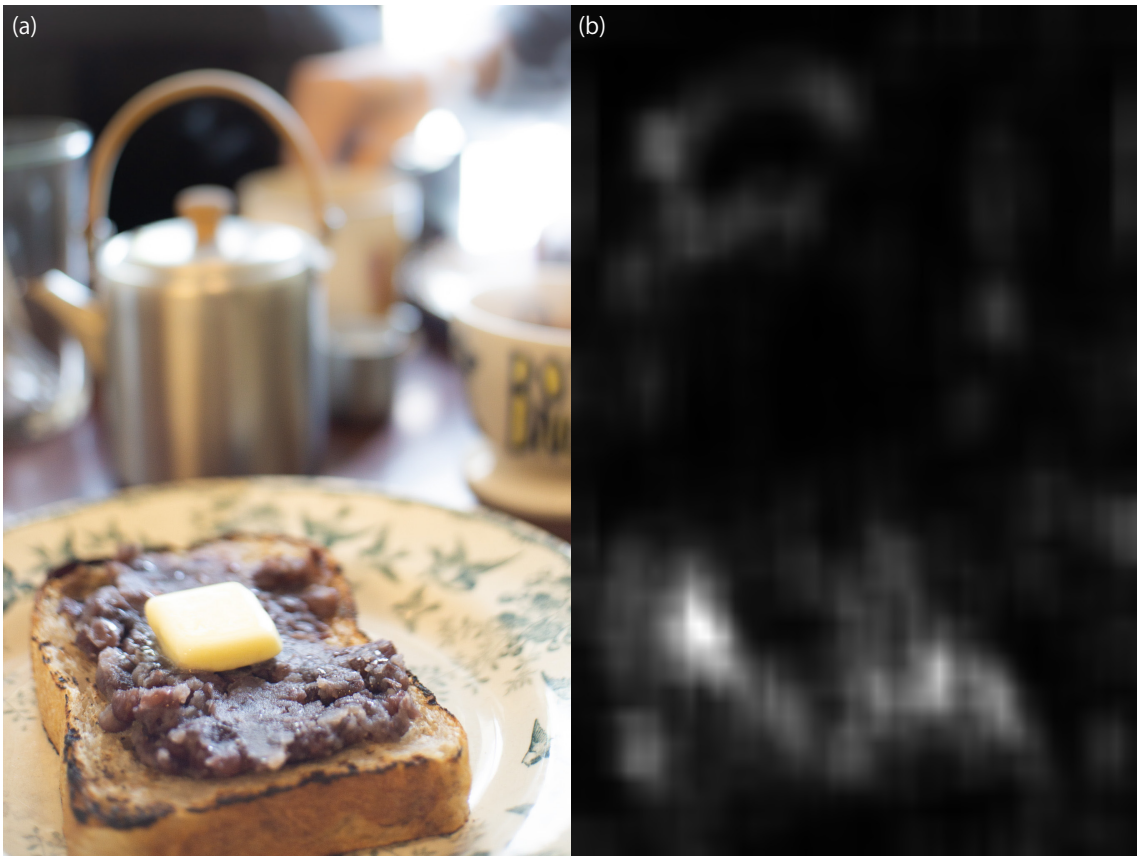


図 2.1: 顕著性マップの生成. (a) 入力画像. (b) 生成された顕著性マップ.

顕著性マップは、画像の中の注目度の高い領域をクロップするために使用されるほか [5, 6], 物体検出や人物検出器としても用いられてきた [7, 8, 9]. 近年では深層学習によって顕著性マップを推定する手法が流行しており, コンピュータビジョン分野の重要な一要素となっている [10, 11].

ビジュアルデザインの分野では, この顕著性マップがテキスト要素とグラフィクス要素の配置関係を推定するために用いられる. 特に雑誌やポスターはある程度その配置関係に決まった型があるという前提のもとで, テキスト要素とグラフィクス要素の配置関係を予め定めたテンプレートを用いる手法が提案されている. Jahanian らは顕著性マップを基にテキスト要素が被らないテンプレートを選択することで, テキスト要素とグラフィクス要素の配置関係を推定した [12]. また, 推定された描画領域内にテキスト要素が収まるよう, テキスト要素の大きさの最適化にも取り組んでいる. 図 2.2 に示すように, Yang らはテキスト要素を雑誌名・ヘッドライン・見出しなどの細かい分類に分けた上でそれぞれ配置を決定した [1]. 加えて, よりグラフィクス要素の特徴を捉えてテキスト要素を配置するべく, 顕著性マップに加えて顔検出や顔方向検出などを組み合わせることでテンプレートの推定に活用した. また, グラフィクス要素の色彩に調和するように, ハーモニックカラー [13] の考えを用いたテキスト要素の色推定を行っている. 筆者もテキスト要素とグラフィクス要素の配置問題に取り組んでいる [14]. 提案した手法では, 物体検出器を用いて具体的に個別の物体の位置を検出し, それを 1 枚の画像に畳み込んだ物体マップを使用することでより確からしく多くの物体が映り込んでいる領域を判定する. これによってグラフィクス要素に明確な被写体が存在する場合, 比較手法と比べてその領域を避けるようにしてテキスト要素の位置を決定することが可能となった. Donovan らはテンプレートを使わずビジュアルデザインの作例から直接レイアウトを学習する手法として, 既存の作例に対して人手でレイアウトを表現するアノテーションデータを作成し, 機械学習を使って配置関係を推定するモデルを提案した [15]. テンプレートを使用する方法はそのシステムが用意した配置関係のみしか表現できない問題があるが, 彼らの手法はその制限を突破できる挑戦的な試みといえる. また, 図 2.3 に示すように彼らはこのレイアウト推定手法を GUI に組み込み, ユーザの編集に対してインタラクティブにレイアウトを変更できるようにしている [16].

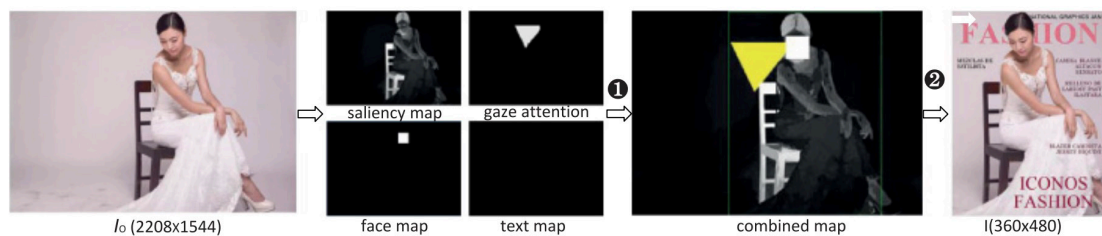


図 2.2: Yang らによるテキスト要素の自動配置 [1].



図 2.3: Donovan らによるユーザの編集に対してインタラクティブにレイアウトを提示するシステム [16]. (図は <http://www.dgp.toronto.edu/donovan/design/index.html> より引用.)

2.2 可読性向上のための読点および改行位置推定

自動で読点・改行を挿入する手法は、主に音声認識の分野において音声認識の結果として得られたテキストを整形するために発展してきた [17]. 英語などのスペースで区切られる言語と異なり、日本語は読点をどこに挿入するかで読みやすさが変化する. 古典的な手法として、鈴木らは既存の文章から句読点が置かれやすい品詞を分析し、形態素解析と独自に作ったルールで句読点を挿入した [18]. しかしながら、読点位置を決定するための共通した一意の規則は存在しないため、ルールベースで行う手法はすべてのパターンを網羅することができない. そこで近年では、村田らにより日本語コーパスから学習した確率モデルで読点を挿入する機械学習的アプローチが提案されている [19, 20]. 彼らは推定のために、形態素・係り受け・節境界などの特徴量を用いた. ここで、同一のテキストであったとしても、アノテーションによって読点を挿入する位置は異なりアノテーションデータに属人性が存在する. 秋田らは読点挿入の属人性の影響を排除するべく、複数人のアノテーションデータを組み合わせることで推定精度を向上させた [21, 22]. 秋田らの研究では、学習したモデルが 78.4% の適合率で読点を推定できることが報告されている.

さらに、音声認識結果を使った応用として、講演での発表者の発話をリアルタイムにスクリーンに表示するユースケースが考えられてきた [2, 23]. この場合、得られた日本語の文章に対して句読点の代わりに改行を挿入することによって読みやすさの向上を狙う. 文章に対して自動で改行を挿入する際も読点と同様に一意の規則は存在しないため、アノテーションデータを用意して機械学習モデルを学習させる. 村田らは係り受けや節境界、音声波形のうち無音部分の時間を表すポーズなどの特徴量をもとに、統計的モデルを使って 80.2% の適合率で改行を推定した. これによってベースライン手法と比較して文章の可読性が上がることを報告している. 大野らはリアルタイムに改行位置を推定する際の手法の違いによる、

手法のリアルタイム性と改行結果の質のトレードオフを比較した [24, 25]. 彼らの実験によると, 文節ごとに改行を挿入する手法が最も定性的評価が高くなることが分かった. 句読点・改行推定のためには形態素解析や係り受け解析の結果などの様々な特徴量を入力とするが, それらの特徴量を計算するための実行時間とその特徴量が改行結果の質に対してどの程度影響しているかという点は今後も考慮すべき項目であると考える. 近年では, Recurrent neural network (RNN) を使って, アノテーションデータに対する推定精度を高める試みがなされている [26, 27].

2.3 E2E なビジュアルデザインの自動生成に向けた周辺研究

2.1 節では, グラフィクス要素とテキスト要素の配置関係に関する研究について取り扱った. グラフィクス要素の自動生成については, コンピュータサイエンスの一大分野である CG や Generative Adversarial Network (GAN) [28] を使ったデータドリブンなグラフィクス生成手法が一般にもよく知られている. 近年では, テキスト要素の自動生成として Natural Language Processing (NLP) の分野で様々な試みがある. Zhang らは, ネット通販サイトに掲載されている実際の商品情報から, キーワードや関連商品のメタデータを取り出し, オートエンコーダで訓練したモデルを用いて広告のキャッチコピーを自動生成した [29]. 加えて, 単なる自動生成にとどまらず, Hughes らは「人間によりクリックされやすい」訴求メッセージを生成することに成功した [30]. これらの研究は, 広告文や訴求メッセージを作るためには専門の知識や経験を積んだクリエイターやコピーライターが必要だという前提を覆し, 自動生成に置き換えることができる可能性を示唆している. 広告のキャッチコピーなどのような短文にとどまらず, コンピュータリソースの増加にともなって小説のような長文の自動生成も可能になっている. Osone らは日本語の Web ページと小説の大規模データセットを用いて学習したモデルにより, 短いキーワード群からタイトルやあらすじを生成した [31]. このように, 既存のデータから学習したモデルによってテキスト要素を自動生成することが可能になりつつあり, 今後はビジュアルデザインにおける応用例も増えていくことが想定される.

2.4 本稿の立ち位置

本稿はビジュアルデザイン分野におけるテキスト要素の自動レイアウトに不足していた要素技術を補完し, 日本語の自動改行技術の分野に対してそれまで主たる関心だった可読性とは別に視覚的バランスという側面に着目した新奇的な挑戦である.

2.3 節で述べたようなグラフィクス要素の自動生成技術とテキスト要素の自動生成技術, そして 2.1 節で述べたグラフィクス要素とテキスト要素の配置問題の研究が進んでいった結果, 雑誌や広告などのビジュアルデザインにおいて人間の手が介入しない E2E なコンテンツ生成が可能になる. これは単純に人手によるコストを省いて効率化・低予算化するだけでなく, 閲覧者のバックグラウンドに合わせてパーソナライズされた雑誌や広告を動的に提示することができ, コンテンツのコンバージョンレートを上げたい制作者にとっても, より

有益な情報を得たい閲覧者にとっても大きな価値がある。一方で、2.1節で述べた自動レイアウトの分野では、テキスト要素の配置や大きさ、色の推定はなされてきたものの、改行については依然として研究の余地がある。なぜなら、テキスト要素をどのように改行・改頁するかで可読性のみならずデザイン全体の印象が変わるからである。本稿では、テキスト要素をどのように分割して表示していくか、そしてそれぞれのテキスト要素内でどのように改行を加えるかを議論し、組合せ最適化問題として定式化する。自動レイアウトの分野においてテキスト要素に適切な改行を加える研究は知られていない。

2.2節で述べたように、従来の文章に対する改行・改頁位置推定では、主に講演で使用する字幕スクリーンへの適用を焦点として可読性のみの最適化に関心があった。これらの研究では、1行あたり20文字程度で設定したアノテーションデータを学習するが、領域が限られ表示できる文字数が少ない場合や、複数のページにまたがって表示される場合、行やページ間で視覚的バランスを取らなければならない場合などについては考慮されていない。また、極端に長さが異なる行が出現することが報告されており、ビジュアルデザインへの応用には課題がある。改行推定研究と比較しても、視覚的バランスを考慮した研究は類をみず、可読性と視覚的バランスの両方を考慮することに新規性がある。

本稿の記述のうち一部は、モバイルコンピューティングとマルチメディア分野の国際会議である MoMM2020 にて発表を行っている [32].

第3章 提案手法

本章では、ビジュアルデザインのための改行・改頁位置推定手法を述べる。まず、3.1節で従来の改行位置とビジュアルデザインにおける改行位置の違いを説明し、本稿で目指すべき改行・改頁について検討する。3.2節では、テキスト要素に対して3.1節で述べた改行・改頁を加えるために、人手を介さずアルゴリズムによって機械的に改行位置を推定する組み合わせ最適化手法について述べる。3.3節では、いくつかの文章に対して提案手法を適用し、改行・改頁を挿入した結果を述べる。

3.1 ビジュアルデザインのための改行・改頁

従来の改行位置推定の研究は2.2節で述べたように、講演での発表者の発話を音声認識しスクリーンに表示するアプリケーションが主であり、可読性のみに焦点が当てられていた。基本的な改行位置推定の考え方としては、図3.1のように文字列の1文字ずつに対して改行可能性 p を推定し、一定のしきい値を越えた位置を改行とみなす。例えば、しきい値 p を0.7と設定したとき、図3.1の例では改行可能性 $p = 0.8$ である「は」の後に改行が加わり、「今日は / 動物園に行ってきました」というスラッシュで改行を表した結果が得られる¹。

今	日	は	動	物	園	に	行	っ	て	き	ま	し	た
$p=0.1$	$p=0.3$	$p=0.8$	$p=0.1$	$p=0.1$	$p=0.3$	$p=0.5$	$p=0.1$	$p=0.2$	$p=0.3$	$p=0.2$	$p=0.1$	$p=0.1$	$p=1.0$

図 3.1: 改行位置推定の基本的な考え方。各文字に対して改行可能性 p を推定し、設定したしきい値を越えた位置に改行を挿入する。

従来研究ではこの改行可能性 p を推定するために、学習用の文章に対して改行位置を示したアノテーションを行い、形態素や係り受け情報などを入力として改行位置を学習するアプローチが取られてきた。本稿で取り扱うビジュアルデザインへの応用では、表示する領域が限られることから行数や文字数に制限が生じるほか、改行の際に可読性以外に考慮すべき点が見受けられる。この節では、そもそもビジュアルデザインにおいてはどのような改行・改頁が望ましいのだろうかという観点で議論を行い、その結論に従って次の節より手法を構築する。

「良い改行とは何か」を議論する際、最も素朴に上げられる指標は可読性（読みやすさ）である。改行のないプレーンテキストに対して、どの位置に改行を付与するかで可読性は

¹本稿では特に断りがない限り“/”で改行を表す

変化する。例えば「今日は動物園に行ってきました」というプレーンテキストに対して「今日は動物園に行/ってきました」と「今日は/動物園に/行ってきました」という2つの改行パターンがあったとき、後者の方が読みやすい。これは、文字列をいくつかの塊に分割して捉えたときに、どの塊の境界に改行が位置するかで説明ができる。塊とは文節や意味合いによって文字列を区切ったときに同じグループに属する文字群のことである。「今日は動物園に行ってきました」の文字列は図3.2のような塊で考えることができる。塊は意味合いによって大きさが異なり、小さな塊を青い領域で表した「今日/は/動物/園/に/行って/き/ました」、中くらいの塊を緑の領域で表した「今日は/動物園に/行ってきました」、大きな塊をピンクの領域で表した「今日は/動物園に行ってきました」のように表現できる。改行を加える際にできるだけ大きな塊の境界に位置するよう挿入することが、可読性を上げるために必要な要素の1つである。

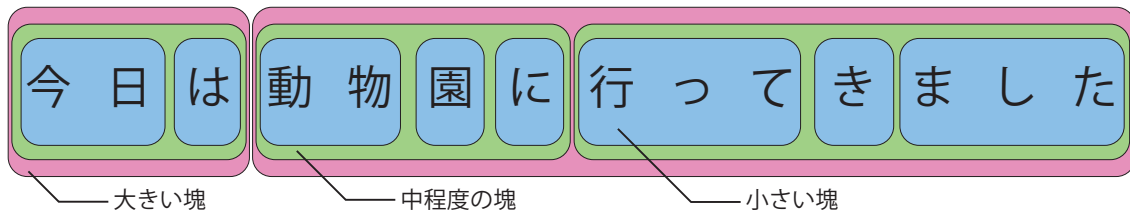


図 3.2: 「今日は動物園に行ってきました」を塊で表した例。

一方で、ビジュアルデザインに応用する際は、必ずしも可読性のみを担保すれば良いわけではない。図3.3に「今日は/動物園に行ってきました」と「今日は動物園に/行ってきました」というテキスト要素をそれぞれグラフィクス要素上にオーバーレイしたときの様子を示す。図3.3 (a) では、文字列の一番大きな塊で改行を加えているため可読性は高いが、グラフィクス要素として目立たせたい被写体（キリン）に被ってしまう。一方で図3.3 (b) では、文字数がちょうど均等になる位置で改行を加えることで被写体に被らないようにテキスト要素が配置されている。テキスト要素だけを見たときに意味合いで改行されているのは図3.3 (a) だが、グラフィクス要素を邪魔しないように改行されているのは図3.3 (b) である。ここで、意味合いによる改行と被写体へのオーバーラップを防ぐことを両立するために、図3.3 (a) のテキスト要素の大きさを小さくするアプローチもある [12]。その場合、グラフィクス要素の顕著性マップを用いて、被写体に被らないようにテキスト要素の大きさを調整する。しかし、テキスト要素はキャッチコピーなどのように閲覧者にメッセージを伝えるよう意図されていることが多く、テキスト要素を小さくすることは望ましくない。また、図3.3 (b) の方がデザイン全体として見たときにバランスが良いという見方もある。これはテキスト要素がメッセージを伝える役割にとどまらず、グラフィクス要素の一要素でもあり全体のデザインに影響を与えるからである。このようにビジュアルデザインでは、テキスト要素を目立たせたい被写体に被らないように配置し、かつそれ自体の視覚的バランスが崩れないようにする必要がある。

これを実現するために、ビジュアルデザインにおける自動改行では文字数や行数に制限

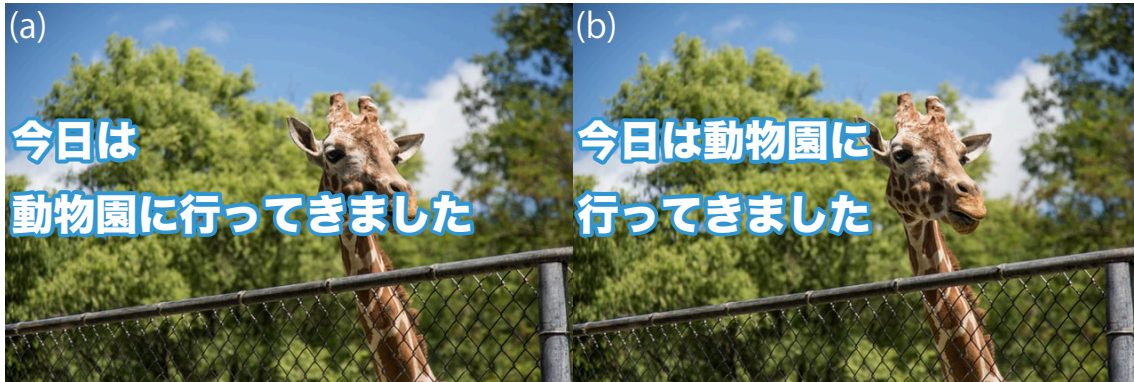


図 3.3: 異なる改行パターンをグラフィクス要素の上にオーバーレイして表示したときの様子.

できるよう設計する必要がある. この設計によって極端に長い行が出現して被写体に重なってしまうことを防ぐことができる. その上で, 行ごとの文字数を均等に近づけることで視覚的バランスを整える必要がある. ただし, 偏った文字数の行を持つテキスト要素を意図的に配置する場合があることに注意する. すなわち通常は視覚的バランスを考慮してテキスト要素に改行を加えるが, デザイナがあえてそれをずらすことによってひと目を引くデザインを実現するテクニックである. 本稿ではこの意図的に視覚的バランスを崩すような改行については検討の対象に含めない.

また, 本稿では可読性と視覚的バランスを以下のように定義する.

- 可読性が高い: 意味合いの塊で区切ったとき, できるだけ大きな塊の境界に改行が位置する.
- 視覚的バランスが良い: 行やページの間で文字数の差が小さい.

さて, 上述の前提のもとでより詳細にパターンを列挙して議論を進める. 例えば「今日は動物園に行ってきました」という文字列を行数2行・1行あたりの文字数無制限という制約の下で改行を加えることを考える. このとき, 文字数は12文字であるので考えられる改行パターンは11通りであり, 図3.4のようなパターンが候補として挙げられる. 横軸に可読性, 縦軸に視覚的バランスを取る. 図3.4 (b) は他の3つのパターンと比較するためにランダムな位置に改行を入れたパターンである. この場合, 意味合いの区切り位置ではない位置に改行が挿入されているため可読性が低いといえ, また行数の分散が大きく視覚的バランスも悪い. 図3.4 (a) は最も視覚的バランスが良くなるように改行を加えたパターンである. これを実現するためには, 1行目と2行目の文字数が近くなる位置に改行を加えれば良い. この場合, 視覚的バランスは良いと言えるが, 意味の塊の間で区切られてしまう可能性があるため可読性は低くなる. 図3.4 (d) は最も可読性が高くなるように改行を加えたパターンである. 最も可読性が高い位置で改行するためには, 各文字の改行可能性 p が最も大きい位置で改行する. この場合, 可読性は高いと言えるが, 文字数は考慮にいけないため視覚的バランスが低くなる可能性がある. 従来研究が目指していた改行推定はこれに該当する. 可読性と視覚的バランスを両立する改行パターンとして, 図3.4 (c) が挙げられ

る。これは文字列の中くらいの塊の位置で改行しているため図 3.4 (a), (b) よりも可読性が高く、図 3.4 (b), (d) よりも視覚的バランスが高いといえる。本稿では、図 3.4 (c) のようなある程度可読性を担保しつつも視覚的バランスの取れた改行位置を推定することを目指す。

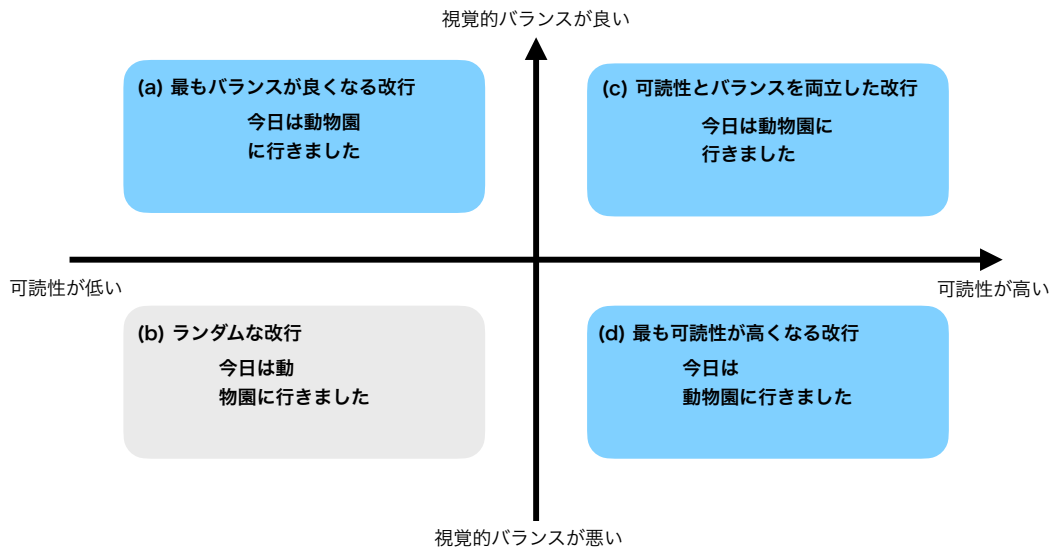


図 3.4: 行数 2 行の制約のもとで考えられる改行パターン。(a) 最もバランスが良くなる改行。(b) ランダムな改行。(c) 可読性とバランスを両立した改行。(d) 最も可読性が高くなる改行。

3.2 改行・改頁位置推定

改行位置を推定するためには与えられた文字列に対して 1 文字ずつ改行可能性 p を推定する。これは文字列を大小からなる塊と捉え、大きな塊の境界ほど高い改行可能性 p を与えることと同義である。文字列を意味の塊で区切るために、これまで形態素解析と係り受け解析が使用されてきた。特に従来研究では、入力として文字列の形態素や係り受け情報、出力として人手でアノテーションされた改行位置を与えることで改行位置を学習・推定している。しかしながら、行数や文字数の制限が動的に変化し、その中で視覚的バランスを考慮する必要のある本稿ではアノテーションは困難である。そこでこの節では、ルールベースによる改行可能性 p の推定と、視覚的バランスを考慮した評価関数による組合せ最適化手法によって改行推定を行う手法を提案する。提案手法は大きく分けて改行可能性推定と組み合わせ最適化の 2 つのフェーズで構成される。3.2.1 項では、ルールベースによって改行可能性 p を推定する簡単な手法を紹介する。3.2.2 項では、可読性と視覚的バランスを両立するための組み合わせ列挙方法と評価関数について述べる。

3.2.1 改行可能性推定

3.1 節で述べたように、本稿では改行可能性 p を正確に求めることよりも、視覚的バランスと両立させることを目指している。そこで、従来手法と比較してより簡易に改行可能性 p を推定するルールベース手法を述べる。

まず、日本語における改行の候補となる位置を考える。例えば、「りんご」という単語があったとき、「り/んご」や「りん/ご」のようにワードの間にスラッシュで示す境界に改行を挿入することはふさわしくない。そのため、最初の手がかりとなるのが形態素である。形態素とは、文字の塊として意味を持つ最小限の単位である。「今日は動物園に行ってきました」という文字列が与えられたとき、形態素解析の結果として表 3.1 が得られる。形態素には、名詞や動詞、助詞などの品詞情報が含まれる。ここで得られる品詞の情報は「形態素解析ツールの品詞体系²」に示されているものを使用した。

M_i	形態素	品詞	品詞細分類
0	今日	名詞	時相名詞
1	は	助詞	副助詞
2	動物	名詞	普通名詞
3	園	名詞	普通名詞
4	に	助詞	格助詞
5	行って	動詞	
6	き	接尾辞	動詞性接尾辞
7	ました	接尾辞	動詞性接尾辞

表 3.1: 形態素解析の結果。

これらの形態素を $M_i (0 \leq i < M)$ と表現する。ここで、 M_0 の形態素は今日、品詞は名詞、品詞細分類は時相名詞である。本稿では形態素解析ツールとして JUMAN++³ を用いた。

さて、ここで「行ってきました」のように 1 つの塊として表現されるであろう文字群が形態素の粒度では「行って」「き」「ました」と分割された状態で表現されている。形態素の中で改行することが望ましくないと同様に、この塊の中も境界にふさわしくないといえる。そこで次にグルーピングする際概念として文節を使うことができる。文節とは、言葉を細かく区切った際に不自然にならない最小の単位で、形態素より大きな粒度となる。日本の初等教育ではしばしば「～ね」を補って区切れるものが分節であると教えられる。「今日は動物園に行ってきました」を例に取ると、「今日はね、動物園にね、行ってきましたね」のように区切れるため、「今日は」「動物園に」「行ってきました」という文節で構成されることが分かる。ところで、形態素解析によって得られた形態素は、大きく分けて自立語と付属語の 2 種類に分けることができる。付属語とは、助動詞と助詞のことで「は」「に」「が」「を」「です」「らしい」などが該当する。自立語とは、名詞や動詞、形容詞などの付属語以

²<https://www.unixuser.org/euske/doc/postag/> (最終閲覧: 2021 年 12 月)

³<http://nlp.ist.i.kyoto-u.ac.jp/index.php> (最終閲覧: 2021 年 12 月)

外の品詞で、「今日」「動物」「行って」「青い」「東京」「変える」「すり代わる」などの例が該当する。文節を構成するルールとして、例外は多く存在するものの、基本的には1個以上の自立語に0個以上の付属語もしくは接辞が接続されたものと定義される⁴。

そこで、形態素を分節にまとめるために、各形態素 M_i に対して以下の条件を満たす位置を分節の区切りとする。

- M_i が接続詞ではない
- かつ、 M_{i+1} が付属語もしくは接尾辞ではない

これを表3.1に適用した結果として、表3.2に示すように「今日は」「動物」「園に」「行ってきました」の文節が得られる。これらの文節を C_j ($0 \leq j < C$) で表すこととする。

C_j	文節
0	今日は
1	動物
2	園に
3	行ってきました

表 3.2: 文節生成の結果.

ここで、この手法により求めた文節は正確な文節ではないことに注意する。「動物園に」という文字列に着目すると「動物」「園に」という結果が得られている。形態素の粒度では、主に哺乳類を表現する名詞である「動物」と、ある一定の目的のために分けられた領域を表現する名詞である「園」は別々の形態素である。しかしながら、先ほどの「～にね」を使って文節を判断する手法を使ったとして「動物のね、園にね」と区切ることは無理があり、「動物園」で1つの複合名詞として扱うことが一般的である。このように、名詞と名詞が繋がっている場合は形態素からのみでは文節を判定できないケースが多い。Wikipediaなどの外部辞書を参照するアプローチも考えられるが、このような固有名詞は文字列として長くなりがちなることも多く、改行パターンを列挙する際に1行あたりの文字数の制限を超えてしまうケースがあるため、本稿では1つの文節で扱わず「動物」「園に」と複数の文節として扱うことにする。今後も本稿で文節と呼ぶ際はこの定義に従う。

最後に改行可能性 p を求める。例えば「～です」「～ます」「～ました」のように、日本語として文章が終了することを示す助詞の直後では強く改行を加えることが望ましい。また、「～に」「～が」のように、現在の境界の直前が付属語で終わっている場合は積極的に改行を挿入したい。一方で、名詞などで終わっている場合は区切りとして弱いことから、改行は避けたい。これらを踏まえて各文字に対して以下のルールを適用して改行可能性 p を決定する。

- その文字が C_j の境界でない場合、 $p = 0.0$ とする。

⁴<https://ja.wikipedia.org/wiki/%E6%96%87%E7%AF%80> (最終閲覧: 2021年12月)

- その文字が C_j の境界で，その C_j の最後の形態素が付属語と接尾辞ではない場合， $p = 0.2$ とする．
- その文字が C_j の境界で，その C_j の最後の形態素が付属語もしくは接尾辞の場合， $p = 0.5$ とする
- その文字が C_j の境界で，その C_j の最後の形態素が動詞性接尾辞か終助詞の場合， $p = 1.0$ とする

動詞性接尾辞には「れる」「させる」などが該当する．終助詞とは，文の終わりに位置する命令・疑問・願望などを意味する助詞のことで，「かしら」「ぞ」などが該当する．

このルールを「今日は動物園に行ってきました」に適用すると，図 3.5 に示す結果が得られる．改行可能性 p は入力した文章に対して文章の長さと同じ大きさを持つ配列の形で得られる．

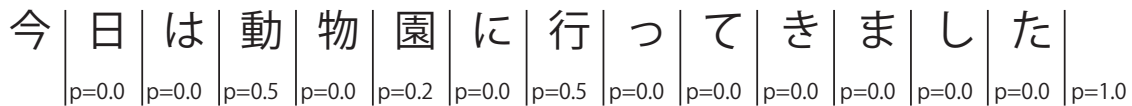


図 3.5: 改行可能性 p の推定結果．

3.2.2 評価関数

3.2.1 項に示した改行可能性 p を用いて改行位置を決定する手法を述べる．単にしきい値を越えたところを改行位置とするのではなく，視覚的バランスを盛り込んだ評価関数を設計し評価値が最も良い改行パターンを採用する．また，動画の字幕やチャットアプリのテキストウィンドウなどのアプリケーションでは改行だけでなく改ページも必要であるため考慮に入れる．

例えば，「美しい写真」という文字列が与えられたとする．ここで改行可能性 p の配列として， $[0.0, 0.0, 0.5, 0.0, 0.0, 0.2]$ が得られる．全ての文字の間に改行・改ページが挿入されるとすると改行・改頁のパターンは 81 通りとなる．しかし， $p = 0.0$ の位置は改行・改頁を挿入されることがないことから，この場合の可能な改行・改頁パターンは，「美しい」と「写真」を単に結合しただけの「美しい写真」，改行記号を挿入した「美しい / 写真」，改ページを加えた「美しい」「写真」の 3 通りとなる．

すなわち，文節として得られた C_j を使って，以下のようなパターンを考慮すれば良い．

1. $C_j + C_{j+1}$
2. $C_j + \text{linefeed} + C_{j+1}$
3. $C_j + \text{pagefeed} + C_{j+1}$

ここで+記号は文字列の結合を示し、linefeed は改行記号の挿入、pagefeed は改ページ記号の挿入処理に置き換えられる。文節の数 C に対して、改行・改頁パターンは $3^{(C-1)}$ 通りとなる。

続いて、それぞれの改行・改頁パターンに対して、可読性と視覚的バランスの定義に基づいてどのパターンが良いかを決定する評価関数を設計する。可読性については、改行・改頁を挿入した直前の文字の改行可能性 p が高いほどよい。そこで、改行・改頁が挿入される直前の文字の改行可能性 p の合計値を S_{sum} とする。視覚的バランスについては、可読性が高い位置にひたすら改行を入れることでページが分散しないよう可能な限り少ないページ数に収めつつ、各行と各ページにおいて文字数の偏りが少ないことが求められる。そこで、 P_{sum} をページ数の総和、 $EP_{textmin}$ を各ページの総文字数の最小値、 $EL_{textmin}$ を各行の総文字数の最小値とする。また、ビジュアルデザインに応用する際に、表示できる領域が限られる、コンテンツによって行数を変えたい、などの要求がある。そこで、 $EL_{textmax}$ を各行の総文字数の最大値、 $EL_{linemax}$ を各頁の総行数の最大値とする。

これらの S_{sum} , P_{sum} , $EP_{textmin}$, $EL_{textmin}$, $EL_{textmax}$, $EL_{linemax}$ を用いて設計した評価関数を式 (3.1) に示す。

$$Eval = -(\alpha \cdot S_{sum} - \beta \cdot P_{sum} + \gamma \cdot EP_{textmin} + \delta \cdot EL_{textmin}), \quad (3.1)$$

$$EL_{textmax} \leq A, EP_{linemax} \leq B$$

$\alpha, \beta, \gamma, \delta$ は係数であり、 $EL_{textmax} \leq A, EP_{linemax} \leq B$ を満たすとする。A, B は制約であり、ユーザあるいは提案手法を組み込むシステムから与えられるアプリケーション用の変数である。

すなわち、行末文字の改行可能性 p が大きく、全体としてページ数が少なく、各行各ページの総文字数の最小値が大きいほど評価関数としては良い値となる。「今日は動物園に行ってきました」に対して制約 A・B をそれぞれ 8・2 としたとき、制約を満たす組み合わせは 5 通りであり、この評価関数による最も評価値の良かったパターンは「今日は動物園に / 行ってきました」であった。これは 3.1 節の図 3.4 で述べた提案手法が目指す改行の要件を満たしている。

3.2.3 最適化

3.2.2 項では本稿の定める改行要件を満たす評価関数を提案したが、この項では評価関数にそって改行パターンを選ぶ際の実装を説明する。

疑似コードをソースコード 3.1 に示す。feeds 変数は文節 C_j を格納する変数であり、MAX_TEXT_COUNT は制約 A, MAX_LINE_LIMIT は制約 B である。nodes は文節と改行、改ページを含んだ改行・改頁パターンを表す。search 関数では、与えられた改行・改頁パターンが全ての文節を含んでいれば評価関数の結果を返す。そうでなければ、その時点での最後のページを取得し、次の文節を追加したと仮定したときに制約 A を満たしていれば

ば文節を追加し再帰する。同様に，次の文節を追加したと仮定したときに制約 B を満たしていれば改行した上で文節を追加し再帰する。また，改ページは必ず可能であるので，改ページして文節を加えたパターンも作成して再帰する。こうして得られた文節を行末に追加したパターン，改行を加えて文節を追加したパターン，改ページを加えて文節を加えたパターンを比較し，最も評価値の良いものを返す。実際には，メモリ領域を効率的に扱うために再帰ではなく for 文による深さ優先探索を使い，また全探索する代わりに簡易な枝刈り用評価関数を使って，それ以上深く探しても良い結果が得られないと判断されれば早急に枝刈りを行う。

```
1 // 改行・改ページアノテーション付きの文節
2 feeds: List
3 MAX_TEXT_COUNT: int = A // 行あたりの文字数制限
4 MAX_LINE_LIMIT: int = B // 行数制限
5
6 void search(nodes: List, i: int) {
7     // 終了条件
8     if i == len(nodes):
9         return evaluate(nodes) // 評価値を返す
10
11     lastPage = getLastPage(nodes) // 最後のページを取得
12     if getLineTextLength(lastPage + feeds[i]) < MAX_TEXT_COUNT {
13         // 末端にテキスト追加
14         result1 = search(nodes + feeds[i], i+1)
15     }
16     if countNewLine(lastPage) + 1 < MAX_LINE_LIMIT {
17         // 改行して追加
18         result2 = search(nodes + [newline] + feeds[i])
19     }
20     // 改ページして追加
21     result3 = search(nodes + [newPage] + feeds[i])
22     return min(result1, result2, result3) // 最小値を返す
23 }
24
25 search(feeds[0], 1) // 探索
```

3.3 結果

提案手法をいくつかの日本語の文章に適用した結果を図 3.6 に示す。“[]”で囲った領域をページ，改行を“\n”で表現した．図 3.6 (a) は制約 B を 1 に，図 3.6 (c) は制約 B を 2 に固定し，それぞれ制約 A を 6 から 20 に変化させたときの結果である．図 3.6 (b) は制約 A を 8 に，図 3.6 (d) は制約 A を 10 に固定し，それぞれ制約 B を 1 から 4 に変化させたときの結果である．どのパターンにおいても読みやすい位置に改行が入っており，また最小のページ数で構成されている．加えて，各行や各ページの文字数が均等になっており資格バランスが良い．例えば，図 3.6 (b), (d) の制約 B が 3 のとき，行数の制約としては 3 行と 1 行の 2 ページで文字列を配置することも可能である．しかし，提案した評価関数によって，2 行ずつページに配置した方が視覚バランスが良いと判断された結果が反映されている．これらの結果より，提案手法による改行・改頁の付与は可読性と視覚的バランスを両立させていることが分かった．

- (a) 今日家族と一緒に動物園に行きました
(Today, I went to the zoo with my family)
- A=6, B=1
[今日家族と] (Today, my family)
[一緒に] (with)
[動物園に] (to the zoo)
[行きました] (I went)
 - A=8, B=1
[今日家族と] (Today, my family)
[一緒に動物園に] (with, to the zoo)
[行きました] (I went)
 - A=10, B=1
[今日家族と一緒に] (Today, with my family)
[動物園に行きました] (I went to the zoo)
 - A=20, B=1
[今日家族と一緒に動物園に行きました]
(Today, I went to the zoo with my family)
- (b) 梅雨前線上に発生した低気圧が西日本に接近します
(A cyclone generated on the Baiu front approaches western Japan)
- A=8, B=1
[梅雨前線上に] (on the Baiu front)
[発生した低気圧が] (A cyclone generated)
[西日本に] (western Japan)
[接近します] (approaches)
 - A=8, B=2
[梅雨前線上に \n 発生した] (on the Baiu front \n generated)
[低気圧が \n 西日本に接近します] (A cyclone \n approaches western Japan)
 - A=8, B=3
[梅雨前線上に \n 発生した] (on the Baiu front \n generated)
[低気圧が \n 西日本に接近します] (A cyclone \n approaches western Japan)
 - A=8, B=4
[梅雨前線上に \n 発生した低気圧が \n 西日本に \n 接近します]
(on the Baiu front \n A cyclone generated \n western Japan \n approaches)
- (c) IT 企業の経営者のための新しい働き方セミナー
(New work style seminar for IT company managers)
- A=6, B=2
[IT 企業の] (IT company)
[経営者の \n ための] (managers \n for)
[新しい働き方 \n セミナー] (New work style \n seminar)
 - A=8, B=2
[IT 企業の \n 経営者のための] (IT company \n for managers)
[新しい働き方 \n セミナー] (New work style \n seminar)
 - A=10, B=2
[IT 企業の \n 経営者のための] (IT company \n for managers)
[新しい働き方 \n セミナー] (New work style \n seminar)
 - A=20, B=2
[IT 企業の経営者のための \n 新しい働き方セミナー]
(for IT company managers \n New work style seminar)
- (d) スキンケアの決定版発売。キャンペーン情報はこちら
(The definitive skin care is released. Here for campaign information)
- A=10, B=1
[スキンケアの] (skin care)
[決定版発売。] (The definitive is released.)
[キャンペーン情報は] (for campaign information)
[こちら] (Here)
 - A=10, B=2
[スキンケアの \n 決定版発売。] (skin care \n The definitive is released)
[キャンペーン情報は \n こちら] (for campaign information \n Here)
 - A=10, B=3
[スキンケアの \n 決定版発売。] (skin care \n The definitive is released)
[キャンペーン情報は \n こちら] (for campaign information \n Here)
 - A=10, B=4
[スキンケアの \n 決定版発売。 \n キャンペーン情報は \n こちら]
(skin care \n The definitive is released \n for campaign information \n Here)

図 3.6: 提案手法を 4 つの文章に適用した結果. (a) 制約 B を固定し, 制約 A を 6 から 20 に変化させたときの結果. (b) 制約 A を固定し, 制約 B を 1 から 4 に変化させたときの結果. (c) 制約 B を固定し, 制約 A を 6 から 20 に変化させたときの結果. (d) 制約 A を固定し, 制約 B を 1 から 4 に変化させたときの結果.

第4章 評価実験

3章では文章の読みやすさとともに視覚的な印象が求められる主としてビジュアルデザインなどへの応用において、読みやすさと視覚的バランスの両方を考慮した改行位置推定手法を提案した。この章では想定するアプリケーションにおける提案手法の有効性を評価するために、クラウドソーシングを用いた定性評価¹を行う。

特に明らかにすべき仮説は次の2点である。

- 仮説(1): 提案手法による文章への改行付与は、ベースライン手法と比較して視覚的バランスを向上させる。
- 仮説(2): 視覚的バランスを考慮することは文章の視認性を向上させ、結果として可読性や理解度の向上に寄与する。

4.1節では仮説(1)に対応する実験として、30文字程度の短文に提案手法によって自動的に改行を加えた際、ベースライン手法と比較して視覚的バランスが向上することを検証する。これは雑誌・ポスター・広告などの画像上にテキスト要素を表示するビジュアルデザインなどのアプリケーションを想定している。続く4.2節では仮説(2)に対応する実験として、メッセージアプリやゲームのチャットウィンドウなどの500文字程度の長文に適用する際に、視覚的バランスを考慮することで可読性や理解度が向上することを検証する。

4.1 短文における視覚的バランスの検証

4.1.1 デザイン

この実験では、提案手法による文章への改行付与がベースライン手法と比較して視覚的バランスを向上させることを検証する。また、ユーザの操作によってインタラクティブにテキスト要素の表示領域が変化し、行あたりの文字数の制限が変わったとしても視覚的バランスは保たれること、視覚的バランスを考慮に入れたことで可読性が落ちていないことを確かめる。

比較対象として、視覚的バランスを考慮せず改行の可能性のある位置でとにかく区切る手法を再現するために、3.2.1項で述べた改行可能性 p の推定の結果を基に、しきい値0.5を越えた位置で改行を加えるベースライン手法を用意した。

実験に使用する文章として、公益財団法人日本国際教育支援協会と独立行政法人国際交流基金が主催する日本語能力試験²より、4段階あるうち上から二番目の難易度に位置する

¹図書館情報メディア系 研究倫理審査 承認第 21-71 号

²<https://www.jlpt.jp/index.html> (最終閲覧: 2021 年 12 月)

N2の問題を用いた。この試験は日本に留学する外国人を対象とした資格試験であるが、その中でもN2は日本人が真面目に問題に取り組めば9割程度の正解率となる程度の難易度である。2012年・2018年発行の公式問題集³の読解問題より、中文問題に該当する文章の中から30文字前後となる文字列を抜き出し、文章が不自然にならないように接続詞等を修正した上で28文字～32文字の範囲に収まるように整形し、これを10セット作成した。読点は読みやすさに関する手がかりを与えてしまうため整形の際に削除した。

今回の実験を実施するために、図4.1に示す実験用のWebサービスを構築した。以下に、実験手順の詳細を示す。

1. 実験参加者は実験の注意文を読んだ後、実験開始ボタンを押す(図4.1(a)).
2. 10セットの文章の中から1つを選び、行あたりの文字数制限を8～20文字の範囲にランダムに決めた上でベースライン手法と提案手法によって自動で改行を加え、その2つのパターンの結果をランダムな順序で左右に並べて提示する(図4.1(b)上)。(ベースライン手法は、文字数の制限を考慮せずとにかく改行を挿入できる箇所区切るため、文字数の制限を変えたときに結果は同一のものとなる。)
3. 読みやすさに関する項目として「読みやすさ(Readability)」, 文章を見た際に受ける印象に関する項目として「視覚的バランス(Visual Balance)」 「自然さ(Naturality)」 「親しみやすさ(Familiarity)」の項目について、2つのパターンのうちどちらがより良いかを「左の方がとても良い」「左の方が良い」「左の方がやや良い」「右の方がやや良い」「右の方が良い」「右の方がとても良い」の6段階の中からそれぞれ選択する(図4.1(b)下)。
4. 10セットの文章がランダムな順序で全て出題され、全ての項目について回答を行った後、実験終了ボタンを押すことで回答をサーバに送信する。

³<https://www.jlpt.jp/samples/sampleindex.html> (最終閲覧: 2021年12月)

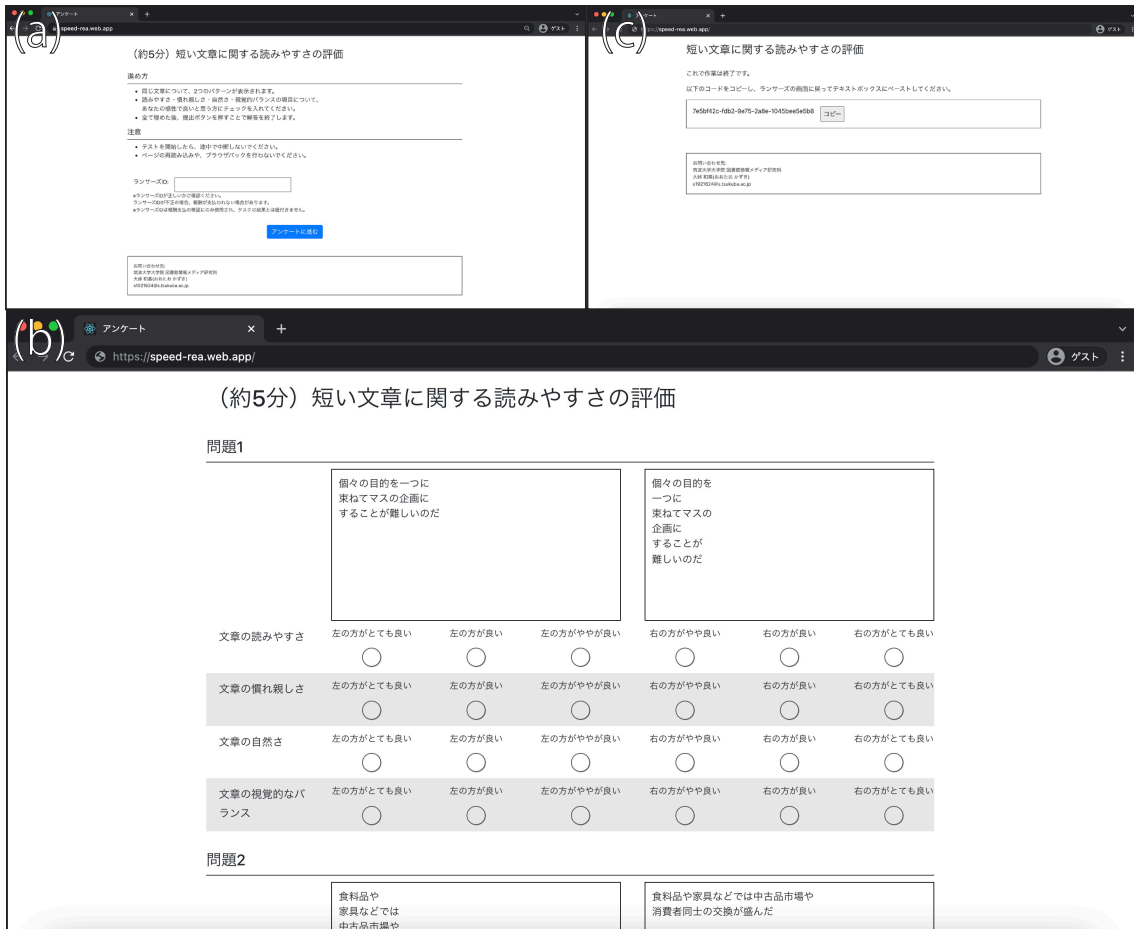


図 4.1: 短文における視覚的バランスの検証用に構築した Web サービス. (a) 説明画面. (b) アンケート画面. (c) 終了画面.

4.1.2 結果

クラウドソーシングサービス「ランサーズ⁴」を用いて実験参加者を集めた。実験参加者は実験に関する注意事項やプライバシーポリシーについて同意したのち、4.1.1項で述べたWebサービスにアクセスし、手順に従って実験に参加した。そのうち207件の回答が得られ、これらを有効な結果として扱った。

得られた6段階評価をバイナリ化した結果を図4.2に示す。実験参加者が回答した6段階評価のうち0から2はベースライン手法、3から5は提案手法として2値化した。読みやすさ、視覚的バランス、自然さ、親しみやすさの全ての項目で提案手法が70%台の評価を得た。特に、自然さと親しみやすさに関しては78%と大きく差がでた。

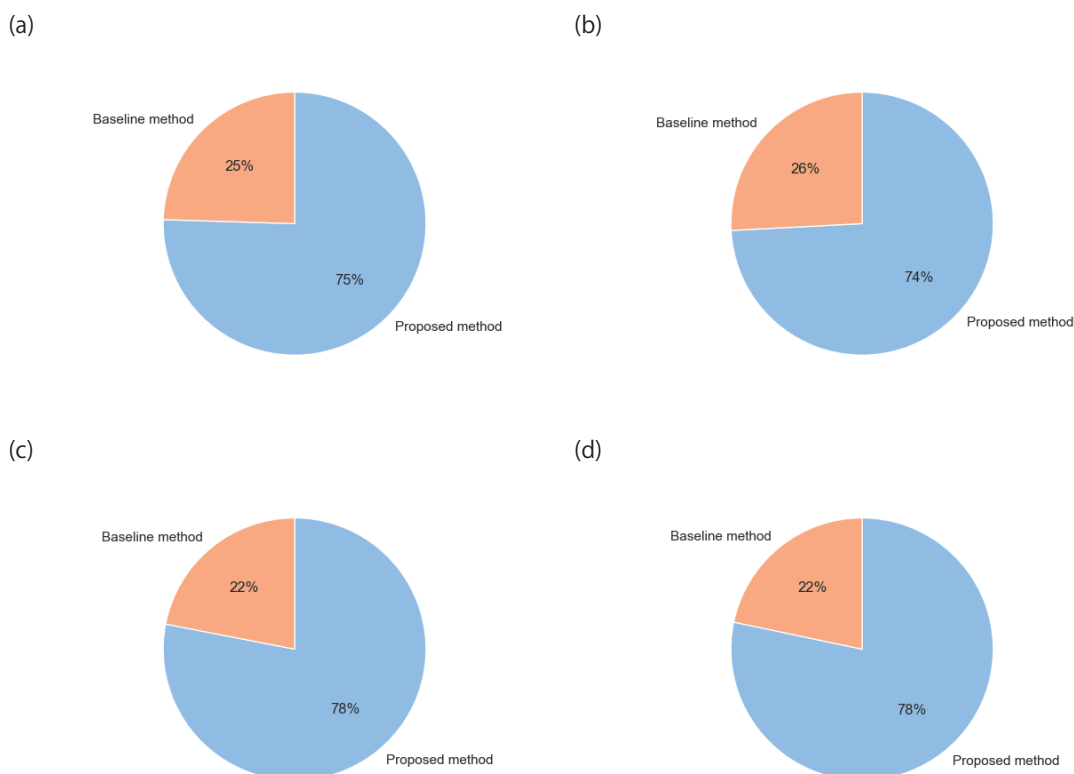


図 4.2: 6段階評価をバイナリ化した結果。(a) 読みやすさ (Readability). (b) 視覚的バランス (Visual Balance). (c) 自然さ (Naturalness). (d) 親しみやすさ (Familiarity).

6段階評価を平均で取ったものを表4.1に示す。評価項目の平均値を取ったものでは、二値化したものと同様、全ての項目で提案手法の方が良いと評価された。また、ベースライン手法と提案手法の中間を示すニュートラル値は2.5であるが、ニュートラル値からの差は0.8程度であった。最も大きく差が出たのは自然さに関する項目であった。

⁴<https://www.lancers.jp/> (最終閲覧: 2021年12月)

評価項目	平均値	ニュートラル値からの差
読みやすさ	3.33	+0.83
視覚的バランス	3.30	+0.80
自然さ	3.36	+0.86
親しみやすさ	3.33	+0.83

表 4.1: 6段階評価の平均値.

4.1.3 考察

この実験では、30文字程度の短文において提案手法の視覚的印象を評価した。結果として、読みやすさ、視覚的バランス、自然さ、親しみやすさの全ての項目でベースライン手法と比較して指標が改善したことを確認した。特に、評価関数として考慮に含まれている読みやすさと視覚的バランスよりも、文章の自然さや親しみやすさの方が上昇幅が大きいことが興味深い。これは、視覚的バランスを考慮することによって、普段の生活でよく見る媒体上での表現に近しくなり、結果的に文章が自然であると判断されたと想定できる。また、6段階評価で表現したときに、ニュートラル値からの差は+0.8程度であった。これは、「提案手法の方がやや良い」と「提案手法の方が良い」の間のうち、やや「提案手法の方が良い」に近い位置にプロットされるものである。このことから、提案手法による読みやすさ、視覚的バランス、自然さ、親しみやすさの向上は劇的なものではなく限定されたものであることに注意する。二値化した結果からも、20%程度の実験参加者がベースライン手法の方が良いと評価していることから、この結果には属人性が存在することにも留意する。

4.2 長文における可読性と理解度の検証

4.2.1 デザイン

4.1 章で述べたように、提案手法による改行は視覚的バランスを向上させるだけでなく、読みやすさにも寄与することが分かった。しかしながら、この結果はキャッチコピーのような 30 文字程度の短文が 1 画面に表示されるというケースに限られたものであり、メッセージアプリやゲームのチャットウィンドウなどの限られた領域にテキスト要素の一部が表示され、ページ送りの要素が加わる中～長文において読みやすさに寄与するかは明らかではない。

そこでこの実験では、500 文字程度の長文を文字数が限られた領域に分割して表示し、ユーザがページ送りをしていくことを想定したときに、機械的に改行・改頁を挿入するベースライン手法と、行やページのバランスを考慮する提案手法を比較する。この実験を実施するための構築した Web サービスを図 4.3 に示す。実験参加者が図 4.3 (a) に示す実験の説明を読んだ後、開始ボタンを押すと、図 4.3 (b) に示すように文章が表示されるテキストウィンドウが提示される。実験参加者はテキストウィンドウ上に表示される文章をできるだけ素早く正確に読むように指示される。この際、図 4.4 のように、実験参加者はテキストウィンドウを操作しページ送りとページ戻しを行うことができる。図 4.3 (b) に示すテキストウィンドウは行あたりの文字数を 20 文字以内、ページあたりの行数を 3 行までに制限されており、この制限を超えてテキスト要素を表示することはできない。テキストウィンドウ上に表示される課題文は、ベースライン手法または提案手法によって自動的に改行・改頁が付与される。

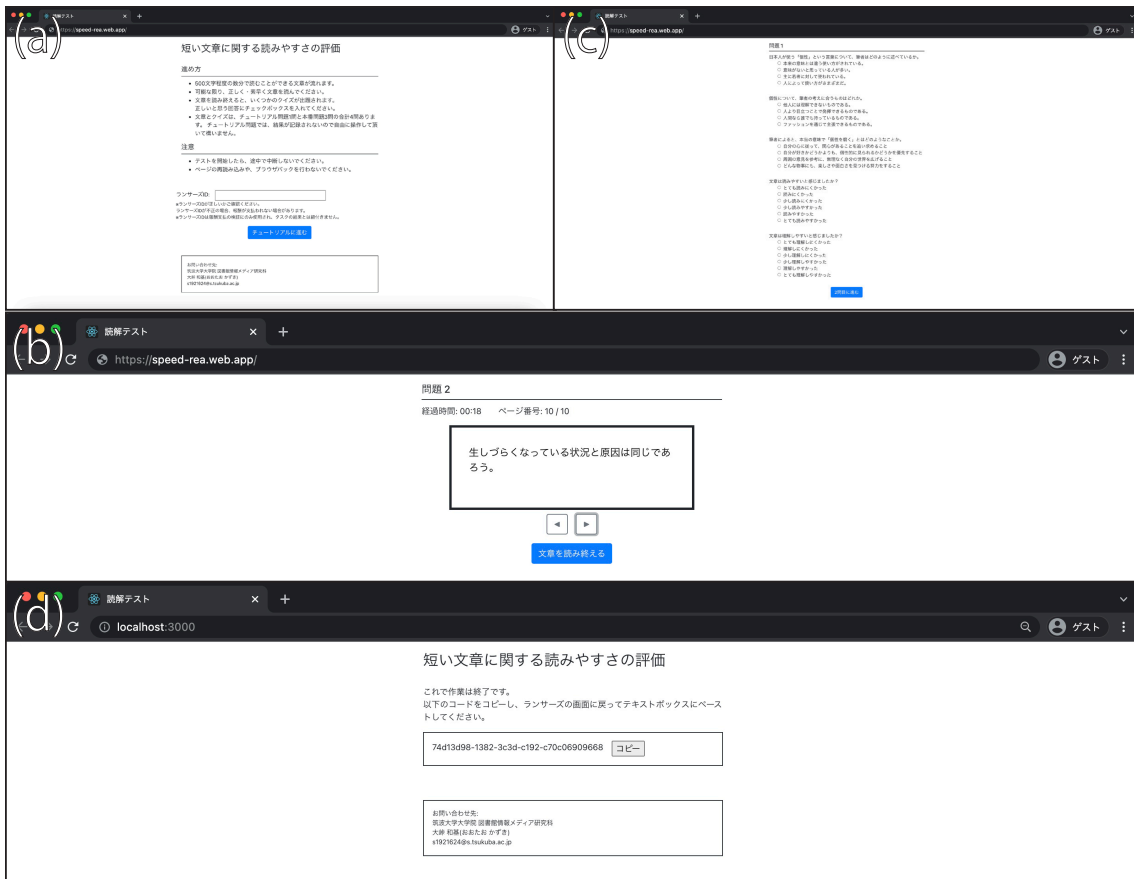


図 4.3: 長文における可読性と理解度の検証のために構築した Web サービス. (a) 説明画面. (b) アンケート画面. (c) 終了画面.



図 4.4: テキストウィンドウの操作. (a) 1 ページ目. (b) ページ送りして 2 ページ目に遷移.

機械的に改行・改頁を挿入するベースライン手法として、

1. ベースライン手法 A: 規定の文字数・行数を超えたときに改行・改頁を挿入する。
2. ベースライン手法 B: 形態素解析で文章を形態素に分解し、規定の文字数・行数の範囲であれば形態素を追加、そうでなければ改行・改頁を挿入する。

の2つの実装を準備した。これらは両方ともナイーブな手法であるが、ベースライン手法 A は意味合いとしてランダムに改行・改頁を挿入したものとみなせるのに対して、ベースライン手法 B は必ず形態素で区切った位置に改行・改頁が付与されるため、ある程度文章の意味合いを反映していることとなる。

実験に使う文章として、実験 1 と同様に日本語能力試験 N2 の 2012 年・2018 年発行公式問題集に記載された読解問題から、中文問題に該当する問題を引用した。実験参加者が問題の全てのページを閲覧したとき、文章を読み終えるボタンが出現し、これ以上読む必要がなければボタンを押すことで図 4.3 (c) に示すクイズ・アンケート画面に遷移する。クイズ・アンケート画面では、課題文に対応する 3 つのクイズと 2 つのアンケートが出題される。クイズの内容は日本語能力試験の公式問題集より引用した課題文に対応する設問であり、4 択のうちから最も回答にふさわしいものを選択する。アンケートとして、「文章は読みやすいと感じましたか?」「文章は理解しやすいと感じましたか?」の 2 つが提示され、「とても読みにくかった/理解しやすかった」「読みにくかった/理解しにくかった」「少し読みにくかった/少し理解しにくかった」「少し読みやすかった/少し理解しやすかった」「読みやすかった/理解しやすかった」「とても読みやすかった/とても理解しやすかった」の 6 択から 1 つを選択する。

問題は全てで 3 問用意されているが、これらの問題に取り組む前に実験参加者は図 4.5 に示すチュートリアル画面を使って、上記で述べたページ送り・ページ戻しや、文章を読み終えるボタン、クイズやアンケートの解答の操作を試す。その後、3 問の問題がランダムな順序で出題され、それぞれ課題文の読解とクイズ・アンケートの解答を行う。ここでは、実験参加者の能力によって特定の改行手法の読解結果にノイズが生じることを防ぐため、同じ実験参加者に対して出す 3 問の問題のうち、適用する改行手法はベースライン手法 A・B・提案手法がランダムな順序で全て適用されるようにプログラムした。全ての問題について回答を行った後、実験終了ボタンを押すことで回答をサーバに送信し、図 4.3 (d) に示す終了画面に遷移する。

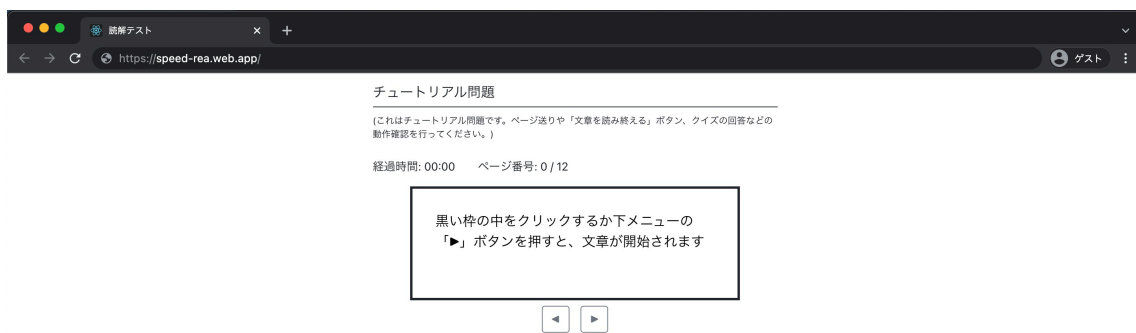


図 4.5: チュートリアル画面.

4.2.2 結果

クラウドソーシングサービス「ランサーズ⁵」を用いて実験参加者を集めた。実験参加者は実験に関する注意事項やプライバシーポリシーについて同意したのち、4.2.1項で述べたWebサービスにアクセスし、手順に従って実験に参加した。実験を通じて235件の回答が得られた。

図4.6に横軸に読解時間と縦軸に正答数を取ったときの実験参加者の結果を青色の丸でプロットした結果を示す。

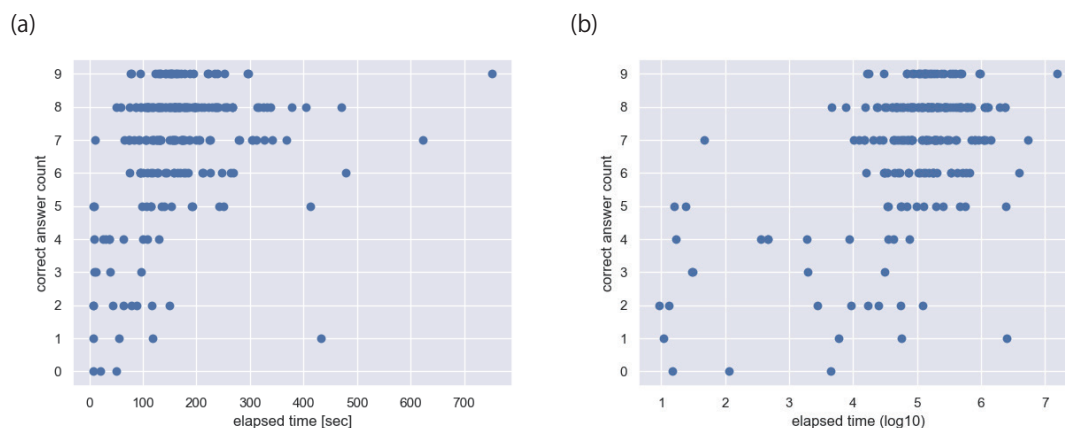


図 4.6: 読解時間と正答数の関係。(a) 読解時間を秒で表現した場合。(b) 読解時間を \log_{10} で表した場合。

読解時間は、実験参加者が課題文を読むのにかかった時間の合計値であり、説明画面やクイズ・アンケート画面、終了画面の閲覧時間は含まれていない。正答数は課題文に対応した3つのクイズの正答数の合計であり、全問不正解で0、全問正解で9の値を取る。図4.6(a)より、多くの実験参加者が合計で300秒以内に課題文を読んでいるのに対して、最長で752秒かかった実験参加者の存在が確認できた。分かりやすさのために読解時間に \log の10底をとったものを図4.6(b)に示す。このとき、読解時間と正答数の相関係数は $r = 0.55$ であり、正の相関にあった。また、上位の正答数を記録している実験参加者は読解時間が4から7の間に位置することが分かる。一方で、1から2に位置する実験参加者は低い正答数となっており、加えてこの時間の範囲で課題文を読むことは非常に困難であるため、報酬目的の不真面目な実験参加者であると言える。そこで、不真面目な実験参加者が結果に及ぼす影響を考慮するため、合計読解時間が3以下の実験参加者を集計から除外することとした。

問題ごとの Word per minute (WPM)、正答数、読解時間の平均を表4.2に示す。

WPMとは、以下の式で表される項目であり、短い時間で正答数が多くなると高くなる。

$$WPM = \frac{1}{\log_{10} elapsed_time} \times \frac{correct_answer_count}{3} \quad (4.1)$$

$elapsed_time$ は読解時間、 $correct_answer_count$ は正答数を示す。WPMは問題A, B,

⁵<https://www.lancers.jp/> (最終閲覧: 2021年12月)

	WPM	正答数	読解時間
問題 A	0.53	2.71	55.48
問題 B	0.48	2.39	55.64
問題 C	0.38	1.96	63.46

表 4.2: 出題した問題の結果.

Cの順に高かった。問題 A, 問題 B の正答数の平均は全問正解した場合の 3 に近い 2.75 と 2.41 であり読解時間も 60 秒を切っていたのに対して, 問題 C では, 正答数の平均が 2 を下回っており読解時間も 60 秒を越えた。このことから, 問題 A・B は比較的簡単な問題であるのに対して, 問題 C は比較的難しい問題であると言える。

問題 A についての, ベースライン手法 A・B および提案手法による違いを表 4.3 に示す。

	WPM	正答数	読解時間	読みやすさ	理解しやすさ	ページ戻し
ベースライン手法 A	0.53	2.68	59.16	4.42	4.58	0.77
ベースライン手法 B	0.53	2.70	54.52	4.55	4.60	0.77
提案手法	0.55	2.74	52.15	4.63	4.82	0.86

表 4.3: 問題 A の結果.

WPM, 正答数, 読解時間, 読みやすさ, 理解しやすさの項目について提案手法が最も良い値だった。それぞれ 2 番目に良い値となったのはベースライン手法 B で, 最下位はベースライン手法 A であった。一方で, ページ戻しの回数は提案手法が最も多かった。ページ戻しの回数は実験参加者がページ戻しを行った回数の平均であり, 全ての手法で 1 を切った。

問題 B についての, ベースライン手法 A・B および提案手法による違いを図 4.4 に示す。

	WPM	正答数	読解時間	読みやすさ	理解しやすさ	ページ戻し
ベースライン手法 A	0.47	2.35	52.53	3.77	3.87	1.41
ベースライン手法 B	0.46	2.33	59.29	3.83	3.82	1.04
提案手法	0.50	2.50	55.25	3.99	3.96	1.57

表 4.4: 問題 B の結果.

WPM, 正答数, 読解時間, 読みやすさ, 理解しやすさの項目について提案手法が最も良い値だった。ベースライン手法の間では, 読みやすさのみがベースライン手法 B の方が良かったものの, WPM, 正答数, 読解時間, 理解しやすさではベースライン手法 A の方が良い値となり, 問題 A と逆転した。ページ戻しの平均値では, ベースライン手法 B が最も少なく, 問題 A と同様に提案手法が最も多かった。

問題 C についての, ベースライン手法 A・B および提案手法による違いを表 4.5 に示す。問題 A・B とは異なり, WPM は 3 つの手法でそれぞれ同程度であり差がでなかった。正答数ではベースライン手法 B と提案手法が同程度であったものの, 読解時間, 読みやすさはベースライン手法 B が最も良かった。そして, 読解時間, 読みやすさ, 理解しやすさの項

目では提案手法が最も悪かった。これらは問題 A・B の傾向と異なっており、また定量的な指標と定性的な指標の傾向が一致しなかった。ページ戻しの回数はベースライン手法 B と提案手法ではほぼ差がでなかったが、ベースライン手法 A がもっとも小さかった。

	WPM	正答数	読解時間	読みやすさ	理解しやすさ	ページ戻し
ベースライン手法 A	0.37	1.91	63.40	3.70	3.66	0.85
ベースライン手法 B	0.38	1.99	61.51	3.72	3.50	1.11
提案手法	0.38	1.98	65.23	3.68	3.46	1.10

表 4.5: 問題 C の結果.

全ての問題をまとめた際の、ベースライン手法 A・B および提案手法による違いを表 4.6 に示す。全体としては、WPM, 正答数, 読解時間, 読みやすさ, 理解しやすさの項目において提案手法が最も良かった。しかしながら、これらの結果に統計的有意差は見受けられなかった。

	WPM	正答数	読解時間	読みやすさ	理解しやすさ	ページ戻し
ベースライン手法 A	0.53	2.68	59.16	4.42	4.58	0.77
ベースライン手法 B	0.53	2.70	54.52	4.55	4.60	0.77
提案手法	0.55	2.74	52.15	4.63	4.82	0.86

表 4.6: 実験全体の結果.

4.2.3 考察

図 4.6 で示したように読解時間と正答数には正の相関があり、文章を読むのに長い時間をかけるほど正答数は上がる傾向が確かめられた。一方で、表 4.6 に示したように、提案手法による改行付与は読解時間を短くしたまま正答数を上げた。これは全体のトレードオフと反しており、限られた領域に表示されるページ送りを伴う長文において、行やページの視覚的バランスを考慮することで読みやすさと理解しやすさを上げることができるといえる。正答数と読解時間の両方を考慮した WPM もわずかであるが高くなっている。加えて、読みやすさと理解しやすさの定性的な評価も提案手法が最も高く、定量的指標の傾向と一致していた。しかしながら、これらの結果の間には統計的有意差は確かめられなかったことには留意する必要がある。

また、難易度が簡単な問題 A・B と、比較的難問である問題 C で傾向が変わることも注意したい。問題が簡単な問題 A・B であれば視覚的バランスを考慮することで結果的に読みやすくなったが、難しい問題である問題 C では視覚的バランスを考慮せず意味合いのみを考慮したベースライン手法 B が優れた結果となった。これは、問題 C は難解な単語が多く文章の接続の関係も理解しづらい構成になっていたことから、下手に見た目のバランスを整えるよりもできるだけテキストウィンドウに表示する文字数が多くなるようにした方が

理解しやすかったのだと推測される。これらの傾向より、視覚的バランスを考慮した改行は簡単な文章であれば理解を助け可読性を向上させる要素になるが、問題が難しければその効果は薄れる傾向にあり、かえって逆効果になってしまうことがいえる。

第5章 応用

提案手法は文章そのものがデザインの一部になりうるビジュアルデザインをはじめとして様々な分野に応用できる。この章では、一例として既にベースの機能が開発されている自動字幕システムとデザイン支援システムに提案手法を組み込むことで応用可能性のデモンストレーションを行う。自動字幕システムでは、音声認識で得られたテキスト要素に装飾をつけて表示する際に、提案手法が持つ「行数や文字数の制約を自由に変えることができる」というメリットを活かして多様な字幕を生成する。グラフィクスデザインツールでは、グラフィクス要素とテキスト要素を自由に配置することができる Web アプリに提案手法を組み込み、テキスト要素に自動で改行を加えることでデザイン制作工程の支援ができることを示す。

5.1 自動字幕システム

5.1.1 背景

音声認識は量子化された音声波形を文字列として変換する処理であり、議事録の文字起こしや聴覚障害のためのリアルタイム字幕システムなど幅広い分野で用いられている。近年では、Google が提供する Google Cloud Speech to Text¹ が句読点の挿入を自動で行うオプションを用意しているが、多くの一般的な音声認識器によって文字列化された発話内容には句読点がない。そこで、2.2 節で述べたように音声認識結果を読みやすく加工するために、講演で用いる聴覚障害のためのリアルタイム字幕システムにおける改行・改頁が探求されてきた。一方で、YouTube 上などに投稿される動画の字幕を自動で生成する際は、読みやすさだけでなく視覚的バランスを考慮する必要があるため提案手法との相性が良い。動画の字幕は単に発話内容を可視化するだけでなく、字幕に装飾やアニメーションを付与することで動画を演出する役割があるからである。そのような応用の場合、字幕自体が持つ視覚的バランスは非常に重要であり、提案手法が解決した問題と一致する。そこで、既に iOS アプリとして開発されている自動字幕アプリに提案手法を組み込むことによって提案手法の応用を示す。

¹<https://cloud.google.com/speech-to-text/> (最終閲覧: 2021 年 12 月)

5.1.2 システム構成

自動で字幕を生成するモバイルアプリケーションに提案手法を組み込むことを検討する。今回は既に iOS アプリとして App Store 上で配布されている自動字幕生成アプリ「Telorain²」に対して提案手法の組み込みを行った。図 5.1 (a) に示すように、従来の処理はユーザがアップロードした音声解析する解析サーバとユーザが操作するモバイルアプリに処理が分けられていた。解析サーバでは動画の音声から発話を検出し、各発話に対して音声認識を行う。その結果を 4.2.1 項のベースライン手法として使ったようなナイーブな手法で改行を加え、装飾テンプレートを適用した。そこで、図 5.1 (b) のような構成で改行を推定するよう変更した。提案手法は改行可能性 p の推定と組み合わせ最適化の 2 つのフェーズに分けられるが、改行可能性 p の推定を解析サーバ上で、組み合わせ最適化をモバイルアプリで行うよう実装した。これは、装飾テンプレートによって行あたりの文字数や行数を変えたいという要求があったこと、そしてモバイルアプリ上で軽量に動作することを両立する構成となっている。

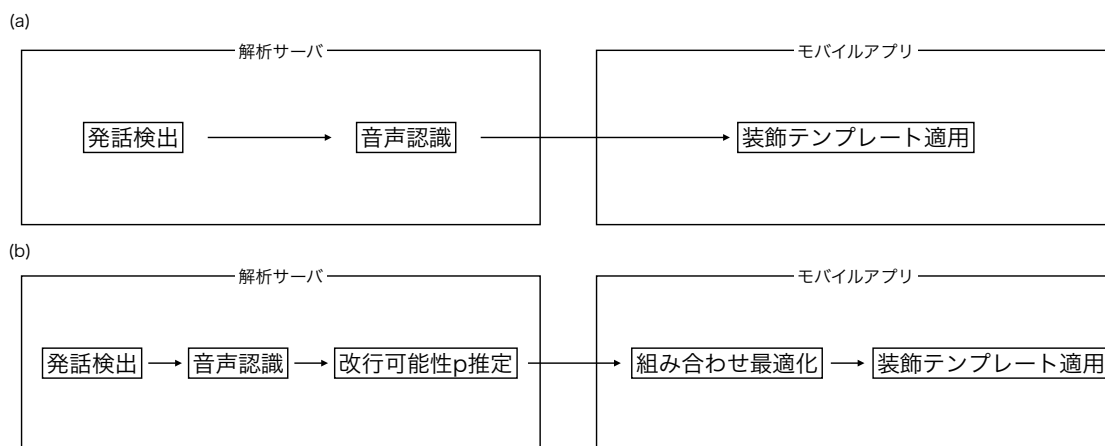


図 5.1: 自動字幕システムの全体像。 (a) 従来の構成。 (b) 提案した構成。

5.1.3 結果

上記の構成を取ることで、可読性が高く視覚的バランスが取れた改行を可能に不仅能够、装飾テンプレートごとに行あたりの文字数と行数の制限を変化させることで多様な字幕を生成できるようになった。図 5.2 に生成された字幕を示す。改行の数を 2 までとすると 2 行の字幕が、3 行とすると 3 行の字幕が構築されるため、装飾テンプレートを作成するデザイナーが改行をデザインの一部として設計できるようになった。組み合わせ最適化の処理はモバイルアプリ上でリアルタイムに行われた。

²<https://apps.apple.com/jp/app/telorain/id1504837403> (最終閲覧: 2021 年 12 月)



図 5.2: 制約を変化させたときの自動字幕生成の様子.

5.2 デザイン支援システム

5.2.1 背景

雑誌や広告などのデザインを制作する際、デザイナーはグラフィクス要素の上にオーバーレイするようにテキスト要素を配置する。その際、グラフィクス要素とテキスト要素の位置関係を決める必要があり、2.1章で述べたように自動で配置関係を推定する研究がなされてきた。ここで、テキスト要素の配置関係だけでなく、テキスト要素にどのように改行を加えるかが視覚的印象のため重要である。そこでグラフィクス要素とテキスト要素を配置できるグラフィクスデザインツールに提案手法を組み込むことによって、デザイン制作の支援が行えることを検証する。

5.2.2 システム構成

大峠と大谷によって提案された、グラフィクス要素とデザイン要素の配置関係を自動で推定する Web ツール [14] に提案手法を組み込む。このツールはグラフィクス要素を読み込んだ状態でテキスト要素を追加すると、自動で適切な位置に配置してくれるものである。テキスト要素に改行を加えるため、新たに3章で述べた手法を実装した API サーバを用意した。この API サーバは HTTP の POST プロトコルで文字列と行あたりの文字数を渡したとき、JSON 形式で改行付きの文章を返すよう設計している。Web フロントエンド側でテキスト要素の文字列か行あたりの文字数が変更されたとき、API サーバにリクエストを送信し、返ってきた文字列をテキスト要素に反映することでテキスト要素への自動改行を実現した。

5.2.3 結果

結果を図 5.3 に示す。左パネルの Text はテキスト要素の文字列、Length Limit は文字数の制限を示す。図 5.3 (a) では Length Limit を制約なしを表現する 0 に指定した。そのため、テキスト要素が被写体である猫に覆い被さるように配置されている。図 5.3 (b) は Length Limit を 40 に設定した。Length Limit を変更すると改行位置の推定が開始され、図のように自動で改行が加わる。これによって被写体である猫と重ならないようにテキスト要素を配置できた。

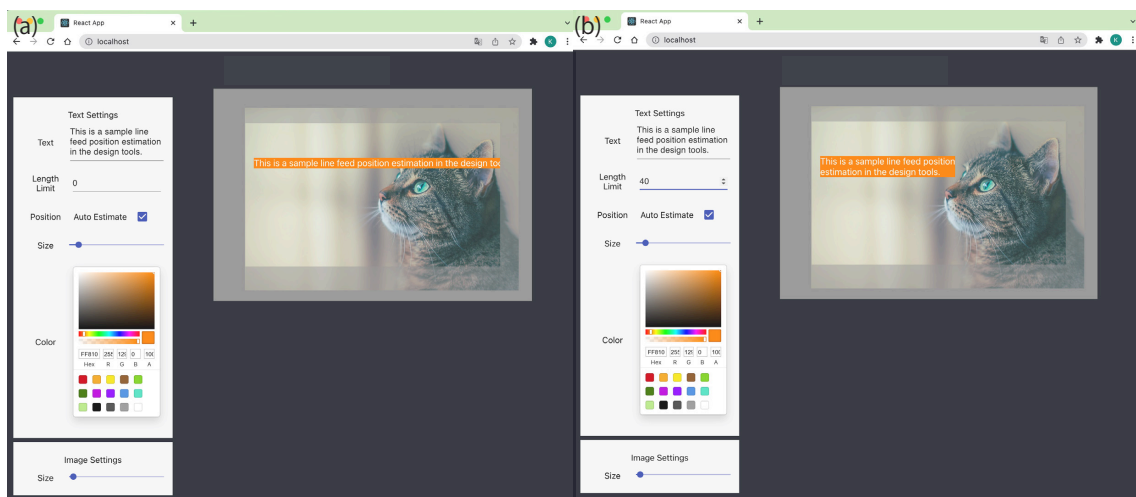


図 5.3: 制約を変化させたときの自動改行の様子.

第6章 議論と今後の展望

6.1 境界に対する改行可能性の推定精度

提案手法では、改行可能性 p を推定するために形態素解析の結果を使用した。具体的には、形態素解析から不完全な文節を推定し、文節の境界に対して $p = 0.2$, $p = 0.5$, $p = 1.0$ を付与した。すなわち、この改行可能性は3段階の強度で表されており可読性の表現に欠ける。本稿の目的は可読性と視覚的バランスの両立であり、この改行可能性 p の推定の精度を上げることが結果に大きく寄与しないと判断したが、従来手法のように係り受け情報などを使うことが改行の質がより向上する可能性がある。例えば、改行可能性 p の推定を他の従来研究や Google が提供する `budou`¹・`budoux`² に差し替えることが可能である。一方で、これらの手法はデータセットに依存するため、本稿のようにデータセットの用意が難しい場合や、手軽にベースライン手法として実装し比較対象にしたい場合は有用であると考えられる。用途に合わせて正しい改行可能性の推定手法を選択することが求められる。

6.2 探索のオーダー

3.2.2 項で述べたように、探索の際は3通りの組み合わせを再帰的に探す。これは $O(n) = 3^{n-1}$ であり、制約 $A \cdot B$ を加えることによって探索空間を抑えられるものの、 n が大きいと容易に組み合わせ爆発を起こす。本稿では応用として雑誌の表紙や広告バナーなどを扱っており、これらの用途ではテキスト要素の文字数が少ないため、大きな問題にはならない。しかしながら、話し言葉のように句点までの文字数が大きくなる文章では、文章を適度に分割した上で提案手法を適用する必要がある。例えば筆者の環境では、 n を10程度に落としたとき携帯端末のアプリ上でリアルタイムに動作した。

より実行速度を重視するには、視覚的バランスを考慮した上で改行可能性 p を推定するアプローチが考えられる。提案手法では、まず改行可能性 p を推定し、その後改行パターンを列挙した上で評価関数の値が最も良いパターンを採用した。しかし、この改行可能性 p を推定するときにそれが視覚的バランスを考慮したものになっていれば、その後の探索フェーズは不要であり単にしきい値処理で済む。ここで問題となるのは、どの境界で改行したかで、他の全ての境界の改行可能性 p が更新されてしまうことである。すなわち、改行可能性 p は、他の改行位置によって動的に変化する。本稿では、この課題に対処するため2段階

¹<https://github.com/google/budou> (最終閲覧: 2021 年 12 月)

²<https://github.com/google/budoux> (最終閲覧: 2021 年 12 月)

からなるフェーズを提案した。さらなる改善については次節で述べるデータドリブンな改行位置推定が考えられる。

6.3 ビジュアルデザインのためのデータドリブンな改行位置推定

本稿では、視覚的バランスを考慮した評価関数を設計したが、3.1節で触れた意図的にバランスを崩した改行は再現できない。これは、グラフィクス要素の構成やテキスト要素のコンテキストに依存するため、これらの事情を反映するのが難しいためである。そこで、ビジュアルデザインの作例から改行アノテーションを作成し、機械学習アプローチによって改行位置を推定する手法が考えられる。テキスト要素のみを考慮に入れた従来の改行位置推定と異なり、グラフィクス要素の顕著性マップなどを入力に組み合わせる必要があり、データセットの準備には更なる工夫が必要となる。本稿では、ビジュアルデザインにおけるテキスト要素の自動レイアウト手法の足がかりとなるように、比較的手法の再現が容易な手法となっている。これをベースライン手法として、今後はデータドリブンな手法の発展に繋げていきたい。

6.4 多言語への適用

この論文では、日本語に焦点を当てて改行手法を論じている。提案手法は改行可能性 p の推定と組合せ最適化の2つのフェーズから構成されるが、英語などの他の言語に提案手法を応用する際は、言語ごとに改行可能性 p を推定する部分を差し替える必要がある。ここで、英語に焦点を当てると、英語圏の文化では読みやすい位置に改行を加えるという文化は見受けられず複数の行にまたがる単語はハイフネーション(-)で接続される。そのため、既存のデータセットから学習することは難しく、人手によるアノテーションは労力がかかる。一方で、英文を読んでいく際に、ある程度まとまった塊に区切ることによって読みやすくする「スラッシュリーディング」と呼ばれるテクニックもある。そこで、本稿でも用いた形態素解析とルールベースによる改行可能性 p の推定に、このスラッシュリーディングのルールを取り込むことができる。スラッシュリーディングには一意のルールは存在せず、人によってやり方はまちまちであるが、一定のルールとして「前置詞」「副詞」「準動詞」「関係詞」「接続詞」などの直前で区切ることが知られている。このように、3.2.1章で述べた手順で形態素解析および境界推定を行うことで、英語への応用は可能である。また、視覚的バランスを考慮する組合せ最適化のフェーズは言語によって変わらないため、日本語に適用した提案手法と全く同じ実装を流用することができる。

第7章 結論

本稿では、ビジュアルデザインにおけるテキスト要素の自動レイアウトの一要素として、文字数と行数に制約を与えた下でテキスト要素に対して自動で改行・改頁を付与する手法を提案した。

従来よりビジュアルデザインの分野ではテキスト要素とグラフィクス要素の自動レイアウト技術が探求されてきたが、特に日本語においては単にテキスト要素を配置するだけでなく、テキスト要素をどのように分割するかで視覚的印象が大きく変化する。また、改行位置を推定する従来研究は、文章の可読性を最適化することに焦点が当てられており、視覚的バランスは考慮されていなかった。本稿では、ビジュアルデザインにおける理想的な改行を議論し、可読性のみならず視覚的バランスを考慮する必要があることを主張した。この視覚的バランスを考慮した改行位置推定を実現するために、(1) 改行可能性 p を推定する簡易なルールベース手法、(2) 可読性と視覚的バランスの両立を目指す評価関数の設計、の2つのフェーズで構成される組合せ最適化を提案した。改行可能性 p の推定では、形態素解析を元に得られた形態素を分節の単位でまとめ、各文節の末尾にある品詞の区分を参照することで、助詞や助動詞などの文章としての区切りを表す部分に高い改行可能性 p を算出した。評価関数の設計では、可読性を表現する項と視覚的バランスを表現する項を含めた評価関数を設計し、組合せ最適化として考えられる改行パターンを列挙することでその評価値の最も良い改行パターンを得た。提案手法をいくつかの文章に適用した結果、提案手法によって改行が付与された文章は可読性と視覚的バランスを両立することを確認した。

加えて、提案手法を評価するためにクラウドソーシングを使ってのべ約400人の実験参加者を集め、2つの実験を実施した。1つ目の実験では、30文字程度の短文に提案手法とベースライン手法を適用し、ベースライン手法と比較して提案手法による改行が文章の「読みやすさ」「視覚的バランス」「自然さ」「親しみやすさ」を向上させることを明らかにした。この改行による印象の改善は劇的ではないものの、70%の実験参加者が提案手法に対してポジティブな反応をみせた。2つ目の実験では、500文字程度の長文に提案手法とベースライン手法を適用し、視覚的バランスを考慮することによって文章の「理解度」と「読解速度」が定量的・定性的の両面で向上することを明らかにした。これは、人間が文章を読むとき、記号としてシーケンシャルに追うのではなく、ある程度文字列を画像として認識しており、視覚的バランスの考慮がその認識プロセスを補助したのだと推測される。また、文章が簡潔なものほど理解度と読解速度の改善効果が大きくなることが分かった。

提案手法は雑誌や広告などのテキスト要素の自動レイアウトに応用できるほか、ブラウザに表示される文字の自動改行、ゲームやチャットアプリ上のテキストウィンドウ、表示で

きる領域が限られた LCD など様々な範囲に応用ができる。本稿では、その一例として自動字幕システムとデザイン支援システムへの応用を試みた。自動字幕システムでは、行あたりの文字数や行数を自由に変えることができるという特徴を活かして、字幕に付与する装飾の表現の幅を広げることができた。デザイン支援システムでは、文字列を入力するだけでテキスト要素の改行と配置が自動で行われ、デザイン制作工程の一部を自動化できることを確認した。

最後に提案手法の制限と今後の展望について議論した。提案手法は、これまでビジュアルデザイン分野で議論されなかった改行によるテキスト要素の自動レイアウト、そして改行推定の分野で議論されなかった視覚的バランスについて言及する野心的な試みであるが、その解決策は依然として改善の余地がある。特に、ルールベースな改行可能性 p の推定と最適化の 2 フェーズについては、得られる改行位置の質と表現の幅に制限がある。例えば、デザイナーが意図的に文字数の偏った行やページを作ることによって表現を行う、グラフィクス要素の顕著性マップやコンテキスト、そしてテキスト要素自体が持つコンテキストを考慮して改行を挿入するようなことはできない。今後はグラフィクス要素の特徴量とテキスト要素の特徴量をともに学習する、マルチモーダルなデータドリブンアプローチの構築を検討していきたい。

謝辞

本研究を進めるにあたって同じ佐藤研究室所属の稲福和史先輩，そして同期にあたる熊田大雅君には，常日頃から研究に関する議論やアドバイスを頂くだけでなく一緒にご飯に行ったり遊びに行くなど公私ともにお世話になりました。また，共同研究室として生活を共にした関研究室の皆様にもお礼を申し上げます。常ににぎやかで楽しい研究室生活は私が研究を進めていく上で心の支えになりました。

指導教員である佐藤哲司先生には，修士課程の途中で研究室を変えることになったにも関わらず快く移籍を認めてくださり，そしていつでも熱心にご指導頂きました。私が無事大学院生としての責務を全うできたのも先生のおかげです。

最後にどんなときでも私の味方となって支えてくれた，父，母，家族の皆様から心から感謝いたします。ありがとうございました。

参考文献

- [1] Xuyong Yang, Tao Mei, Ying-Qing Xu, Yong Rui, and Shipeng Li. Automatic generation of visual-textual presentation layout. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Vol. 12, No. 2, pp. 1–22, 2016.
- [2] Masaki Murata, Tomohiro Ohno, and Shigeki Matsubara. Construction of linefeed insertion rules for lecture transcript and their evaluation. *International Journal of Knowledge and Web Intelligence*, Vol. 1, No. 3-4, pp. 227–242, 2010.
- [3] Simon Lok and Steven Feiner. A survey of automated layout techniques for information presentations. *SmartGraphics*, Vol. 2001, pp. 61–68, 2001.
- [4] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, pp. 1–8. Ieee, 2007.
- [5] Edoardo Ardizzone, Alessandro Bruno, and Giuseppe Mazzola. Saliency based image cropping. In *International Conference on Image Analysis and Processing*, pp. 773–782. Springer, 2013.
- [6] Ziaur Rahman, Yi-Fei Pu, Muhammad Aamir, and Farhan Ullah. A framework for fast automatic image cropping based on deep saliency map detection and gaussian filter. *International Journal of Computers and Applications*, Vol. 41, No. 3, pp. 207–217, 2019.
- [7] Sebastian Montabone and Alvaro Soto. Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image and Vision Computing*, Vol. 28, No. 3, pp. 391–402, 2010.
- [8] Jingfan Guo, Tongwei Ren, and Jia Bei. Salient object detection for rgb-d image via saliency evolution. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2016.
- [9] Jianjun Lei, Bingren Wang, Yuming Fang, Weisi Lin, Patrick Le Callet, Nam Ling, and Chunping Hou. A universal framework for salient object detection. *IEEE Transactions on Multimedia*, Vol. 18, No. 9, pp. 1783–1795, 2016.
- [10] Mohammed Brahimi, Marko Arsenovic, Sohaib Laraba, Srdjan Sladojevic, Kamel Boukhalfa, and Abdelouhab Moussaoui. Deep learning for plant diseases: detection

- and saliency map visualisation. In *Human and machine learning*, pp. 93–117. Springer, 2018.
- [11] Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. Evaluating saliency map explanations for convolutional neural networks: a user study. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pp. 275–285, 2020.
- [12] Ali Jahanian, Jerry Liu, Qian Lin, Daniel Tretter, Eamonn O’Brien-Strain, Seungyon Claire Lee, Nic Lyons, and Jan Allebach. Recommendation system for automatic design of magazine covers. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pp. 95–106, 2013.
- [13] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. In *ACM SIGGRAPH 2006 Papers*, pp. 624–630. 2006.
- [14] 大峠和基, 大谷まゆ. バナー制作のための背景を考慮した自動テキスト配置. 人工知能学会全国大会論文集 第 34 回全国大会 (2020), pp. 3Rin485–3Rin485. 一般社団法人人工知能学会, 2020.
- [15] Peter O’Donovan, Aseem Agarwala, and Aaron Hertzmann. Learning layouts for single-pagegraphic designs. *IEEE transactions on visualization and computer graphics*, Vol. 20, No. 8, pp. 1200–1213, 2014.
- [16] Peter O’Donovan, Aseem Agarwala, and Aaron Hertzmann. Designscape: Design with interactive layout suggestions. *ACM Conference on Human Factors in Computing Systems*, pp. 1221–1224, 2015.
- [17] Ji-Hwan Kim and Philip C Woodland. The use of prosody in a combined system for punctuation generation and speech recognition. In *Seventh European conference on speech communication and technology*, 2001.
- [18] 鈴木英二, 島田静雄, 近藤邦雄, 佐藤尚. 日本語文章における句読点自動最適配置. 全国大会講演論文集, データ処理, pp. 185–186, 1995.
- [19] Masaki Murata, Tomohiro Ohno, and Shigeki Matsubara. Automatic comma insertion for japanese text generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 892–901, 2010.
- [20] 村田匡輝, 大野誠寛, 松原茂樹. 読点の用法的分類に基づく自動読点挿入. 研究報告音声言語情報処理 (SLP), Vol. 2010, No. 8, pp. 1–8, 2010.
- [21] Yuya Akita and Tatsuya Kawahara. Automatic comma insertion of lecture transcripts based on multiple annotations. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

- [22] 秋田祐哉, 河原達也. 講演に対する読点の複数アノテーションに基づく自動挿入. 情報処理学会論文誌, Vol. 54, No. 2, pp. 463–470, 2013.
- [23] 村田匡輝, 大野誠寛, 松原茂樹. 読みやすい字幕生成のための講演テキストへの改行挿入. 電子情報通信学会論文誌 D, Vol. 92, No. 9, pp. 1621–1631, 2009.
- [24] Tomohiro Ohno, Masaki Murata, and Shigeki Matsubara. Sequential linefeed insertion into lecture transcriptions for real-time captioning. *Electronics and Communications in Japan*, Vol. 98, No. 2, pp. 20–31, 2015.
- [25] 大野誠寛, 村田匡輝, 松原茂樹. 講演のリアルタイム字幕生成のための逐次的な改行挿入. 電気学会論文誌 C (電子・情報・システム部門誌), Vol. 133, No. 2, pp. 418–426, 2013.
- [26] 飯泉智朗, 大野誠寛, 松原茂樹. 読みやすい字幕生成のための rnn を用いた講演テキストへの改行挿入. 第 82 回全国大会講演論文集, Vol. 2020, No. 1, pp. 451–452, 2020.
- [27] 飯泉智朗, 大野誠寛, 松原茂樹. 読みやすい字幕生成のための講演テキストに対する改行挿入の改善. In *IEICE Conferences Archives*. The Institute of Electronics, Information and Communication Engineers, 2020.
- [28] Liang Gonog and Yimin Zhou. A review: Generative adversarial networks. In *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 505–510. IEEE, 2019.
- [29] Yuchi Zhang, Yongliang Wang, Liping Zhang, Zhiqiang Zhang, and Kun Gai. Improve diverse text generation by self labeling conditional variational auto encoder. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2767–2771. IEEE, 2019.
- [30] Hughes John, Keng-Hao Chang, and Ruofei Zhang. Generating better search engine text advertisements with deep reinforcement learning. In *Proceedings of KDD*, 2019.
- [31] Hiroyuki Ozone, Jun-Li Lu, and Yoichi Ochiai. Buncho: Ai supported story co-creation via unsupervised multitask learning to increase writers’ creativity in japanese. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–10, 2021.
- [32] Kazuki Otao and Tetsuji Satoh. Text layout methods overlaid on images considering readability and balance. In *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia*, pp. 219–223, 2020.