

平成 2 8 年度卒業論文

高速逐次視覚提示された文章の  
最適表示時間と快適速度

情報・通信工学専攻 コンピュータサイエンスコース

1 3 1 1 1 6 3 肥後 亮佑

指導教員 寺田 実 准教授

提出日 2 0 1 7 年 1 月 3 1 日

# 概要

## 目的

電光掲示板で文章が表示されるときには文章全体は表示されず横スクロールで表示されることが多い。また、スマートウォッチなどの画面が小さい端末で長い文章を読むこと容易ではない。

高速逐次視覚提示という方法では一定の場所にテキストが次々と表示されていくので、視点を固定したまま文章を読むことができる。この方法で文章を表示する単位は英語の場合は単語が多い。日本語の場合は意味的区切りである文節で区切って表示するのが最適だと考えられる。

本論文では、日本語の文章を文節で区切って高速逐次視覚提示した場合に固定長で区切った文と比べてどれだけ快適に読める速度が向上するのかを調べる。

## 方法

まず予備実験を行い、日本語の文節に関する情報と文節の表示時間にどのくらいの相関関係があるかを調べる。その後、予備実験によって最も相関関係のある文節に関する情報を用いて表示時間を最適化して表示し、快適に読める速度に調節してもらう。5文字で区切って自動で表示する方法と比較して、快適速度がどれだけ向上しているかを調べる。

## 結論

表示時間を最適化した方法のほうが快適速度が速かった人が4名、変化なしの人が1名、遅かった人が1名となった。

# 目次

第 1 章	序論	6
1.1	背景	6
1.2	現状の問題	6
1.3	目的	6
1.4	本論文の構成	6
第 2 章	関連研究	8
2.1	文章を読むときの眼球の運動	8
2.1.1	停留とサッカード	8
2.1.2	サッカードが不要な場合の読み速度	8
2.1.3	RSVP で表示する単語と表示時間の関係	9
2.1.4	サッカードが不要な読みの利点	9
2.2	日本語の場合	9
2.2.1	日本語で RSVP を適用する場合の区切り	9
2.2.2	ページ送りと改行がない電子ブックリーダー・インターフェース	10
2.2.3	横スクロール速度の快適速度	11
第 3 章	提案手法	13
3.1	概要	13
3.2	文節の表示時間に影響を及ぼす要素	13
3.3	予備実験	14
3.3.1	概要	14
3.3.2	実験方法	14
3.3.3	実験結果	14
第 4 章	実装	18
4.1	開発環境	18
4.2	処理の流れ	18
4.3	文節区切り	19
4.4	最適表示時間の決定	20
4.5	自動表示	20

目次	3
第5章 評価実験	21
5.1 概要	21
5.2 実験方法	21
5.3 実験結果	22
第6章 考察	23
6.1 予備実験に関する考察	23
6.1.1 文節の難易度	23
6.1.2 漢字の影響	24
6.2 評価実験に関する考察	24
6.2.1 自動表示の効果	24
6.2.2 快適速度	24
6.3 応用例	25
第7章 結論	26
7.1 まとめ	26
7.2 今後の課題	26
参考文献	28

## 目次

2.1	停留とサッカードによる視点の動き <sup>*1</sup> . . . . .	8
2.2	RSVP の例 . . . . .	9
2.3	表示時間と単語の長さの関係 [2] . . . . .	10
2.4	固定長改行レイアウト (上) と文節間改行レイアウト (下)[3] . . . . .	11
2.5	ページ送りと改行がない電子ブックリーダー・インターフェース [4] . . . . .	11
2.6	分かち書き無し (上) と分かち書き有り (下) の表示 [5] . . . . .	12
3.1	予備実験の画面構成 . . . . .	15
4.1	本論文で使ったアプリケーション . . . . .	18
6.1	表示した文節の難易度別分布 . . . . .	23

## 表目次

3.1	被験者毎 t 値 . . . . .	15
3.2	重回帰分析の結果 . . . . .	16
3.3	要因を文節の文字数だけにした時の回帰分析の結果 . . . . .	16
3.4	推定読み速度 . . . . .	17
3.5	被験者全員のデータをまとめた回帰分析の結果 . . . . .	17
5.1	2つの方式の特徴 . . . . .	21
5.2	RSVP における快適速度 . . . . .	22
6.1	予備実験に使用した文節の難易度 . . . . .	23
6.2	予備実験と評価実験の読み速度 . . . . .	24

# 第 1 章 序論

## 1.1 背景

電光掲示板で表示される文章は表示範囲の制限により横スクロールで表示されることが多い。スマートウォッチなど小さな画面では長い文章を読むことは容易ではない。このように表示範囲が限られている場合に文章を読む時には通常の文章の表示方法（固定長改行レイアウト）で表示するのは困難である。

今後、デジタル携帯端末がさらに小型され、IoTなどでさまざまなものに画面や通信機能やデジタルの要素が組み込まれるようになることを考えると、小さい画面で多くの情報を素早く得ることは有用であると考えられる。

## 1.2 現状の問題

英語などの単語と単語の間が分かち書き、すなわちスペースで区切られている言語では、高速逐次視覚提示という方法で単語を順番に表示していくことで表示範囲が限られていたとしても文章を読むことができる。日本語は分かち書きされる言語ではないので文章をどこかで分割する必要があるが、その単位としては意味的な区切りである文節が良いと考えられる。

日本語を文節に区切ることができても表示時間をどう決定するのかという問題が生じる。日本語は漢字、平仮名、片仮名などが含まれていてそれぞれの役割や一文字あたりの情報量は異なる。したがって、文節ごとに最適な表示時間が異なると考えられる。

## 1.3 目的

本論文では日本語の文章を文節で分割し、高速逐次視覚提示で表示して読む場合の表示速度を最適化して、実際に速く、快適に読むことができたかを評価することを目的とする。そのためにまず予備実験を行い、人が日本語の文章の文節を読む速度を計測し、その結果から日本語の文章の文節を読む時間に影響する要素を決定して回帰式を求め、この回帰式を利用して高速逐次視覚提示で文節を表示するときの表示時間を最適化する。

## 1.4 本論文の構成

論文の構成を簡単に説明する。本章では、序論として研究の背景と問題点、目的について述べた。第 2 章では、関連研究について述べる。第 3 章では、提案手法について述べる。第 4 章では、システムの実装について述べる。第 5 章では、評価実験について述べる。第 6 章では、予備実験と評価実験に対する考察について述

べる．第7章では，まとめと今後の課題について述べる．



## 第 2 章 関連研究

### 2.1 文章を読むときの眼球の運動

#### 2.1.1 停留とサッカード

人間は文章を読むときの眼球の運動はなめらかな動きではなく停留とサッカードを繰り返している。サッカードとは停留点から次の停留点へ素早く視点が移動する運動である。

図 2.1 は停留とサッカードの例である。視点が赤色の矢印の先までサッカードで移動し、その点で停留するという運動を繰り返す。行の右端まで行くと次の行の先頭までサッカードで戻る。

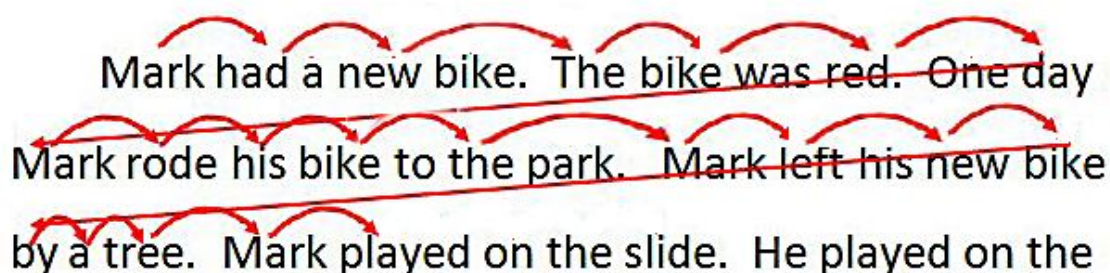


図 2.1: 停留とサッカードによる視点の動き\*1

#### 2.1.2 サッカードが不要な場合の読み速度

サッカードが不要な文章の表示をすれば文章を速く読めるのではないかという観点から、文章の表示に高速逐次視覚提示 (Rapid Serial Visual Presentation) という表示方法が用いられる場合がある。RSVP は画像や文字などを連続的に高速で表示するものである。図 2.2 は RSVP を用いて一単語ごと連続で表示することで文章の速読が可能となる Spritz\*2 というソフトウェアの画面である。この図では "Welcome" という単語が表示されたあとに "to" という単語が表示されたことを表している。

Rubin & Turano(1992)[1] は RSVP で文章を表示したあとに選択式問題で内容理解の問題を出すという実験を繰り返して、正解率が 75 % 以上のままでどのくらい読み速度を上げることができるのかを求めた。この結果、ある被験者は約 2 倍の速度で文章を読めたことが示された。

\*1 <http://eyecanlearn.com/tracking/>

\*2 <http://spritzinc.com/>



図 2.2: RSVP の例

### 2.1.3 RSVP で表示する単語と表示時間の関係

Rubin & Turano(1992) の研究では単語は同じ時間間隔で表示されていたが、ESP(Elicited Sequential Presentation)[2] という方法では被験者がボタンを押して次の単語を表示するという方法で文章を読ませた。これにより RSVP より 47 %速く文章を読めることが示された。さらに、単語の長さ并表示時間の間に相関係数 0.285 の相関関係があることを示した [2]。この結果から RSVP で表示される単語の表示時間は表示される単語の長さによって変えたほうがより速く読めるということが分かる。図 2.3 は縦軸に単語の長さ、横軸に表示時間をとったグラフである。また、回帰分析の結果次の式が得られた。

$$\text{表示時間} = 80.7 * \text{単語の長さ} + 495.8(\text{ms}) \quad (2.1)$$

### 2.1.4 サッカーが不要な読みの利点

サッカーが不要な文章の読みの利点として文章が速く読めるということ以外にも、電光掲示板やスマートウォッチなどの限られた範囲で文章を表示するときや、視点を動かすことが出来ないロービジョンの人たちが文章を読むときに利用できると思われる。

## 2.2 日本語の場合

### 2.2.1 日本語で RSVP を適用する場合の区切り

日本語は英語とは異なり、単語と単語がスペースで区切られていない。したがって、日本語で RSVP を行う場合には日本語をどこで区切ればいいのかという問題が生じる。

小林らは文章を意味的な区切りである文節で改行して表示する電子リーダーを提案した [3]。一般的な文章

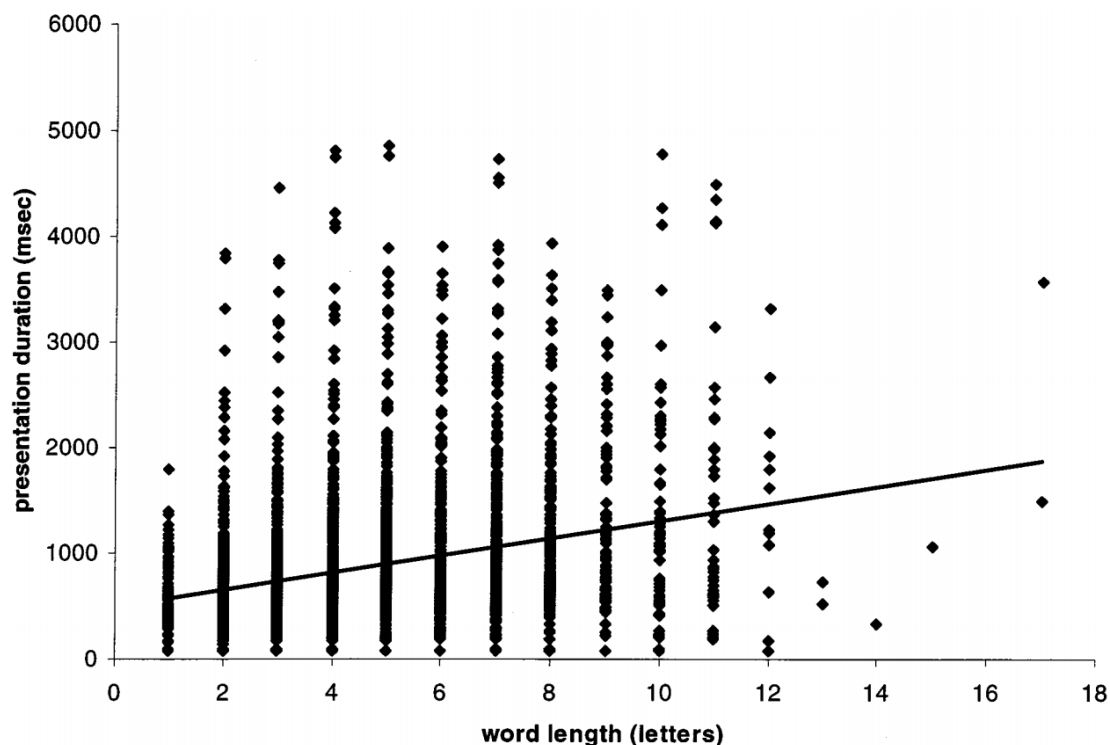


図 2.3: 表示時間と単語の長さの関係 [2]

表示レイアウトとして固定長改行レイアウトというものがあるが、これは改行位置が一行の長さによって決まり、意味的な区切りではないところで改行されてしまう可能性がある。これにより読者は改行位置での文章理解が遅くなり、読み速度が低下する。この電子リーダーでは意味的な区切りである文節で改行されるので改行による文節の分断による意味の理解が遅くなることなく、この電子リーダーでは読み手が表示された文章を上下にスクロールする必要があるが、固定長改行レイアウトと比べて読み速度が 29 % 向上したことを示した。図 2.4 の上が固定長改行レイアウトで、下が文節間改行レイアウトである。

この研究から、日本語の文章を区切るときの単位は文節が最適であり、RSVP で日本語を表示するときの区切りとしても文節を使用したほうがよいと考えられる。

### 2.2.2 ページ送りと改行がない電子ブックリーダー・インターフェース

川崎、中田はページ送りと改行がない電子ブックリーダー・インターフェースを開発した [4]。このリーダーでは RSVP で日本語の文章を分割して表示している、この分割された文章の表示時間は文字列の長さによって決まっており、長い文字列の方が長く表示される。

図 2.5 はこのブックリーダーの画面であり、“吾輩は猫である。”という文を分割して“吾輩は”と“猫である。”に分割して表示している。

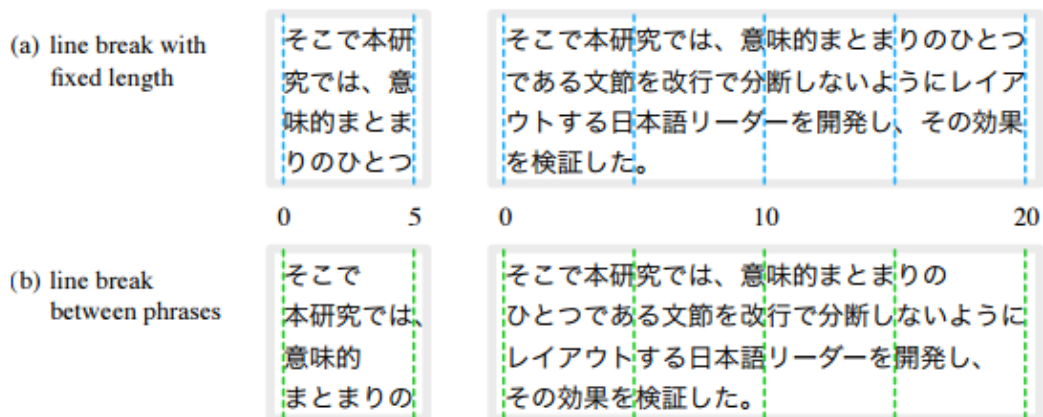


図 2.4: 固定長改行レイアウト (上) と文節間改行レイアウト (下)[3]

下の表示方法で 29 %読み速度が向上した .



図 2.5: ページ送りと改行がない電子ブックリーダー・インターフェース [4]

### 2.2.3 横スクロール速度の快適速度

水口, 岡本は 1 行スクロールを基本とする電子ビューアを提案し, 快適速度を求める実験を行った [5]. 実験では, 表示する文章を分かち書きした場合としない場合のときの快適速度を求めた. 実験の結果, 表示する文章を分かち書きした場合としない場合での快適速度に一定の法則が見られなかった.

図 2.6 の上は横スクロールにおいて分かち書き無しで文章を表示した場合であり, 下は分かち書き有りて文章を表示した場合である.

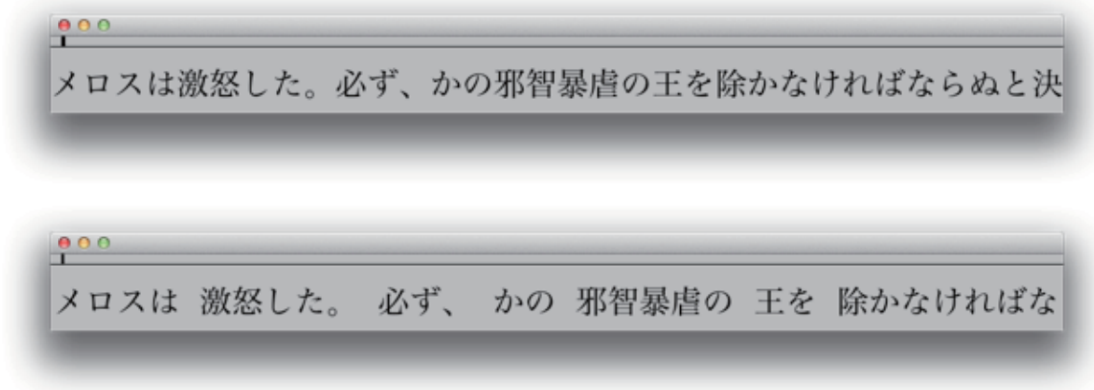


図 2.6: 分かち書き無し(上)と分かち書き有り(下)の表示 [5]

## 第3章 提案手法

### 3.1 概要

本論文では日本語の文章を最小の文節単位で分割して、その文節ごとの表示時間を最適化して高速逐次視覚提示する手法を提案する。文節によって最適な表示時間は異なると考えられるので、文節の文字数、難易度、漢字がどれだけ含まれているかなどの要素と表示時間との相関関係がどれだけあるかを予備実験によって調べ、その結果に基づいて最適な表示時間を決定する。

### 3.2 文節の表示時間に影響を及ぼす要素

文節の表示時間に影響を及ぼすと考えられる要素としては次のような要素が考えられる。

- 文節の文字数
- 文節に含まれる漢字の数
- 文節の難易度

これらは文節の文字数が長いとそれだけ情報量が多くなり、漢字は平仮名、片仮名とくらべて一文字あたりの情報量が多く、文節の難易度が高ければ理解に時間がかかるという仮定に基づいている。

文節の難易度は日本語教育語彙表 ver1.0<sup>\*1</sup>に含まれる単語の難易度を利用した。この表は海外の日本語学習者のために作られ、日本語の教師の主観で難易度が付与されている。この表には約 18000 語の単語が収録されており、それぞれの単語に初級前半、初級後半、中級前半、中級後半、上級前半、上級後半という難易度が割り当てられている。この順番で 1~6 の番号を割り振り、番号が大きい方が難易度が高いようにした。文節から助詞、助動詞、接尾辞、数字、記号を除外したときに残る単語に含まれる単語の中で最も高い難易度を文節の難易度とした。文節に辞書に載っていない単語があった場合その文節の難易度は 7 とした。残った単語がなかった場合にはその文節の難易度は 0 とした。

文節の文字数と文節に含まれる漢字の数という要素は独立していないので、次の 3 つの要素にした。

- 文節の文字数 – 文節に含まれる漢字の数
- 文節に含まれる漢字の数
- 文節の難易度

---

\*1 <http://jhlee.sakura.ne.jp/JEL.html>

この3つの要素は独立していると考えられるので、それぞれを変数  $X, Y, Z$  とすると、表示時間  $T$  は次の回帰式で表せる。 $a, b, c, d$  は被験者によって異なる値をとる。

$$T = aX + bY + cZ + d \quad (3.1)$$

## 3.3 予備実験

### 3.3.1 概要

回帰式 3.1 で表現される表示時間のモデルが実際に成り立つかどうか調べるために予備実験を行う。

文節を自動で表示していると、読者が表示されている文節を読めないまま次の文節が表示されてしまう可能性があるため、この実験では被験者がボタンを押して次の文節を表示するようにした。これによって、文節を読めないまま次の文節を表示することがなくなり、文節を読むのにどれだけ時間がかかっているかを測定することができる。ボタンを押す動作によって表示時間が長くなると想定されるが、この方法で影響が出る変数は切片の  $d$  だけだと考えられる。

### 3.3.2 実験方法

本実験では NHK NEWS WEB<sup>\*2</sup> の記事の約 10 万件の第一パラグラフのテキストを文節で区切って表示して、被験者がボタンを押して次の文節を表示するという方法で表示された文節と表示時間の情報を調べる。

実験に使用したアプリケーションの画面は図 3.1 に示した。区切られた文節は画面上部に表示される。被験者は右下の Random というボタンを押して 10 万件のニュースの記事の中から 1 つをランダムで表示する。スペースキーを押すことで次の文節を表示することができる。最後の文節まで表示すると画面上部に何も表示されなくなる。再び Random ボタンを押すとランダムでニュースが表示される。図 3.1 では“成田山新勝寺で、”という文節が表示されている。被験者が自分のペースでスペースキーを押してニュースを読む。表示するテキストサイズは 44px とした。

実験で得られるデータは以下の 5 つである。

- 表示した文節の文字列
- 文節の表示時間
- 文節の文字数
- 文節に含まれる漢字の数
- 文節の難易度

文節の表示時間とは、文節が表示されてからスペースキーが押されるまでの時間であり、単位はミリ秒とする。

### 3.3.3 実験結果

被験者は 20 代の男性 5 名と女性 1 名の合計 6 名であり、速読などを意識せずに自分のペースでニュースを読むように指示した。合計約 350 個の文節を表示してもらった。得られたデータから明らかな外れ値を除く

<sup>\*2</sup> <http://www3.nhk.or.jp/news/>

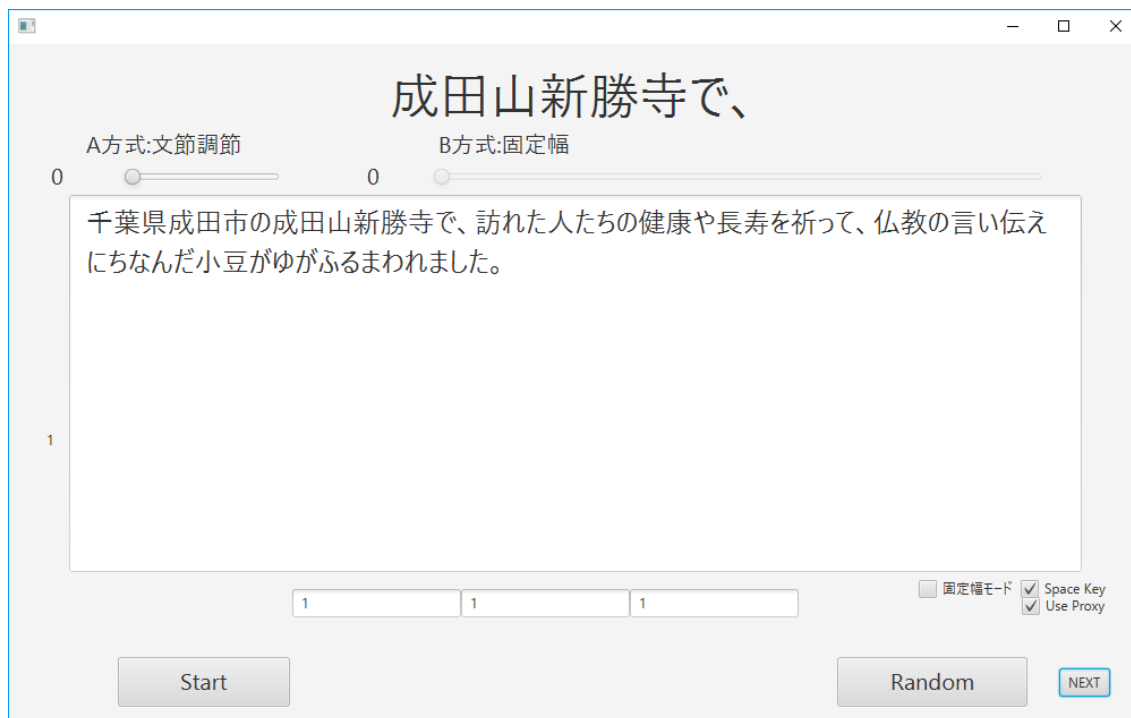


図 3.1: 予備実験の画面構成

て、残ったデータの中の最新の 300 件を利用して重回帰分析を行ったところ次の結果が得られた。重回帰分析には Excel 2016 のアドインの分析ツールを用いた。

表 3.1: 被験者毎 t 値

t 値	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
被験者 A	3.40	3.43	-0.93	19.4
被験者 B	5.15	13.8	-1.54	4.43
被験者 C	5.15	3.14	0.262	18.8
被験者 D	6.76	18.3	0.128	4.38
被験者 E	5.63	9.22	-0.183	23.2
被験者 F	2.10	5.36	-1.31	9.76

一般に回帰分析でそれぞれの変数の係数の t 値の絶対値が 2 以下であるならばその変数は意味のない変数となる。したがって回帰分析の結果より文節の難易度を表す変数 *Z* の係数 *c* の t 値の絶対値はすべての被験者で 2 以下となっているので、文節の表示時間を説明する変数として *Z* は不要であると考えられる。この原因についての考察は第 6 章で述べる。これにより、3.1 から変数 *Z* を除くことで次の式が得られる。

$$T = aX + bY + d \quad (3.2)$$

この回帰式に基づいて再び被験者 A ~ F のデータに対して重回帰分析を行うと表 3.2 の結果が得られた。



重相関  $R$  とは回帰式の変数が2つ以上のときの相関係数であり、式 3.2 では左辺の値と右辺の値の相関係数を表す。重決定  $R^2$  は決定係数とも呼ばれ、回帰式によって左辺のデータがどれくらい説明できているかということを表す指標である。この値は回帰分析において最小二乗法を用いた場合は重相関  $R$  の2乗と一致する。本実験で重回帰分析に使用した分析ツールは最小二乗法を用いている。補正  $R^2$  とは自由度調節済みの決定係数とも呼ばれ、重決定  $R^2$  の値を回帰式の変数の数の影響を考慮して調節したものである。この値は  $0 \leq R^2 \leq 1$  という条件を満たし、この値が大きければモデルの当てはまりが良いということが分かる。一般的に補正  $R^2$  の値を重視する。

表 3.2: 重回帰分析の結果

被験者	重相関 $R$	重決定 $R^2$	補正 $R^2$	$a$	$b$	$d$
A	0.225	0.0506	0.0442	7.60	9.66	245
B	0.632	0.399	0.395	48.8	174	178
C	0.315	0.0995	0.0934	13.6	12.0	296
D	0.747	0.557	0.554	54.3	160	181
E	0.501	0.251	0.246	30.2	67.8	665
F	0.295	0.0873	0.0811	11.3	41.2	316

また、表示時間の要因として文節の文字数のみを用いて回帰分析した場合の結果を表 3.3 に示した。

表 3.3: 要因を文節の文字数だけにした時の回帰分析の結果

被験者	重相関 $R$	重決定 $R^2$	補正 $R^2$
A	0.222	0.0491	0.0459
B	0.471	0.222	0.219
C	0.315	0.0989	0.0959
D	0.608	0.369	0.367
E	0.418	0.174	0.172
F	0.195	0.0380	0.0348

これにより、文節の表示時間の要因として文節の文字数を用いるモデルより、[文節の文字数-文節に含まれる漢字の数] と [文節に含まれる漢字の数] という2つの要因とするモデルにするほうが補正  $R^2$  が高くなっている被験者は4名で、低くなった被験者は2名となった。この低くなった被験者の補正  $R^2$  の差に有意差は認められなかった。したがって、この被験者は漢字とその他の要素の影響には差がないことが分かる。この原因については第6章で述べる。得られた回帰式から推定読み速度(文字/分)を求めた結果は表 3.4 のようになった。

以上は、被験者ごとの分析であったが、被験者6人のデータをすべてまとめて3.2の回帰式で回帰分析すると以下の結果が得られた。

これより被験者を平均化した次の回帰式が得られた。

$$\text{表示時間} = 31.3 * (\text{文節の文字数} - \text{文節に含まれる漢字の数}) + 85.8 * \text{文節に含まれる漢字の数} + 351 \quad (\text{ms}) \quad (3.3)$$

表 3.4: 推定読み速度

被験者	読み速度 (文字/分)
A	989
B	428
C	790
D	434
E	319
F	657

表 3.5: 被験者全員のデータをまとめた回帰分析の結果

重相関 $R$	重決定 $R^2$	補正 $R^2$	$a$	$b$	$d$
0.387	0.150	0.149	31.3	85.8	351

## 第4章 実装

### 4.1 開発環境

本システムは JavaFX を用いて実装した。JDK の準拠は JavaSE-1.8 である。開発環境は Eclipse<sup>\*1</sup> を用いた。UI の実装は Scene Builder 2.0<sup>\*2</sup> を用いた。

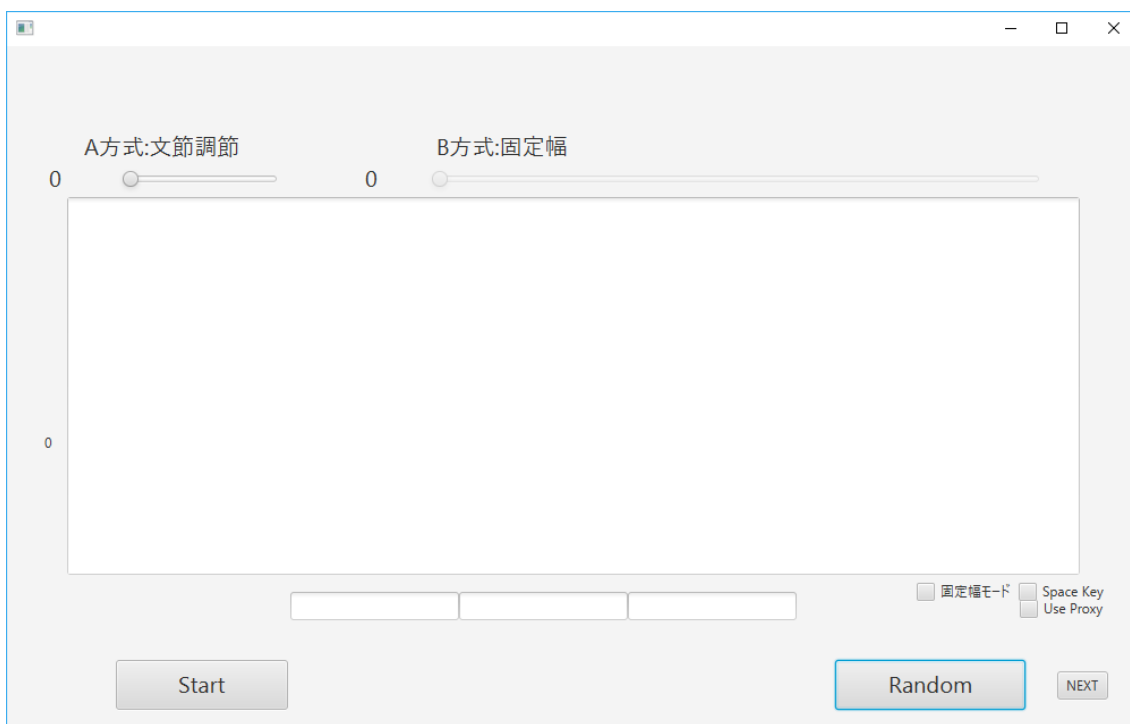


図 4.1: 本論文で使用したアプリケーション

### 4.2 処理の流れ

実験の流れは以下のようになる。

1. 式 3.2 のパラメータ  $a, b, d$  を入力

<sup>\*1</sup> <https://eclipse.org/>

<sup>\*2</sup> <http://gluonhq.com/labs/scene-builder/>

2. 読みたい文章を入力
3. Web API により文節で区切る
4. 予備実験によって得られた回帰式より表示時間を決定
5. 文章を自動で表示

### 4.3 文節区切り

日本語テキストを文節に区切るのに Yahoo ディベロッパーネットワークのテキスト解析 Web API の日本語係り受け解析<sup>\*3</sup>を用いた。Web API へ GET リクエストを送って、レスポンスとして得ることができる xml ファイルの Chunk というフィールドの中 Surface というフィールドの中の単語を結合したものを文節とする。

例としてソースコード 4.1 の場合を考える。これは“うちの庭には二羽鶏がいます。”というリクエストに対するレスポンスとして得られた xml の一部である、この場合 Chunk というフィールドの中に Surface というフィールドがある。ここに形態素解析された文字が含まれているのでこの Surface フィールドの中身を結合すると“うちの”が得られる。これを文節とする。次の Chunk フィールドにも同様の処理をすると“庭には”という文節が得られる。

```
<Chunk>
  <Id>0</Id>
  <Dependency>1</Dependency>
  <MorphemList>
    <Morphem>
      <Surface>うち</Surface>
      <Reading>うち</Reading>
      <Baseform>うち</Baseform>
      <POS>名詞</POS>
      <Feature>名詞,地名町名,*,うち,うち,うち</Feature>
    </Morphem>
    <Morphem>
      <Surface>の</Surface>
      <Reading>の</Reading>
      <Baseform>の</Baseform>
      <POS>助詞</POS>
      <Feature>助詞,助詞連体化,*,の,の,の</Feature>
    </Morphem>
  </MorphemList>
</Chunk>
<Chunk>
  <Id>1</Id>
  <Dependency>3</Dependency>
  <MorphemList>
    <Morphem>
      <Surface>庭</Surface>
      <Reading>にわ</Reading>
      <Baseform>庭</Baseform>
      <POS>名詞</POS>
      <Feature>名詞,名詞場所,*,庭,にわ,庭</Feature>
    </Morphem>
    <Morphem>
      <Surface>に</Surface>
      <Reading>に</Reading>
      <Baseform>に</Baseform>
      <POS>助詞</POS>
      <Feature>助詞,格助詞,*,に,に,に</Feature>
    </Morphem>
    <Morphem>
      <Surface>は</Surface>
      <Reading>は</Reading>
      <Baseform>は</Baseform>
      <POS>助詞</POS>
      <Feature>助詞,係助詞,*,は,は,は</Feature>
    </Morphem>
  </MorphemList>
</Chunk>
```

ソースコード 4.1: “うちの庭には二羽鶏がいます。”というリクエストに対するレスポンスとし

<sup>\*3</sup> <http://developer.yahoo.co.jp/webapi/jlp/da/v1/parse.html>

て得られた xml の一部

## 4.4 最適表示時間の決定

文節が得られると回帰式から最適表示時間を計算することが出来る。

例として被験者 A が“IT企業グーグルは、”という文節を表示することを考える。この文節の文字数は 10 であり、漢字の数は 2 であるから、回帰式に基づいて表示時間は

$$T = a * (10 - 2) + 2b + d \quad (4.1)$$

となり、この式に被験者 A のパラメータを代入すると

$$T = 7.60 * (10 - 2) + 2 * 9.66 + 245 \quad (4.2)$$

$$= 325(\text{ms}) \quad (4.3)$$

となる。

## 4.5 自動表示

回帰式を用いること表示する全ての文節の表示時間を決定することができるので、この表示時間で文節を順番に表示していく。文節の表示時間が経つと次の文節が表示される。

ソースコード 4.2 では自動表示のソースコードの一部を表している、変数 `displayTime` には回帰式から計算された表示時間 (ms) が格納されている。表示時間を制御するのに `Thread.sleep()` を用いている。

```
try {
    Thread.sleep(displayTime);
} catch (InterruptedException e) {
    e.printStackTrace();
}
```

ソースコード 4.2: 自動表示

## 第 5 章 評価実験

### 5.1 概要

予備実験によって得られた回帰式に基づき被験者ごとに最適な表示時間で RSVP を行うが、予備実験では文節を見てスペースを押すという動作が入っており、自動で表示される RSVP とは最適な表示時間に僅かな差があると考え、回帰式の切片を被験者に調節してもらうという方法で快適速度を求める。比較対象として、固定の文字数を一定の時間間隔で RSVP するという方法でも快適速度を求めた。

### 5.2 実験方法

まず実験に使用する 2 つの表示方式の特徴を表 5.1 にまとめた。

表 5.1: 2 つの方式の特徴

	表示文字数	表示時間
文節最適方式	文節の文字数	回帰式で最適化した時間
固定長方式	5 文字	一定

文節最適方式とは予備実験から得られた回帰式を用いて表示する文節の表示時間を最適化する方式であり、固定長方式とは文章を 5 文字で区切って同じ時間間隔で表示する方式である。

評価実験に使用したアプリケーションの画面は図 4.1 に示した。実験の流れは以下のようにした。

1. 式 3.2 のパラメータ  $a, b, d$  を画面下のテキストフィールドに入力
2. Random ボタンを押す
3. 文節最適方式で表示されている文章を読みながら自分が快適だと感じる速度にスライダーを調節
4. 2~3 を快適なときのスライダーの値に変化が無くなるまで繰り返す
5. モードを切り替えるために右下の固定長のチェックボックスにチェックを入れる
6. Random ボタンを押す
7. 固定長で表示されている文章を読みながら自分が快適だと感じる速度にスライダーを調節
8. 6~7 を快適なときのスライダーの値に変化が無くなるまで繰り返す
9. モードを切り替えるために右下の固定長のチェックボックスにチェックを外す
10. 快適速度に変化が無くなるまで 2~9 繰り返す

その後、調節されたスライダーの値を報告してもらった。

### 5.3 実験結果

被験者は予備実験に参加した6名である。調節してもらったスライダーの値から、快適速度を求めた結果を以下の表にまとめた。この表から、一分間で快適に読むことができる文字数が分かる。

表 5.2: RSVP における快適速度

被験者	文節最適方式 (文字/分)	固定長方式 (文字/分)
A	1200	1200
B	586	353
C	2640	3000
D	513	500
E	385	375
F	743	500

4名の被験者が文節最適方式のほうが快適速度が速く、1名は固定長方式のほうが速く、1名は同じ速度であった。

## 第6章 考察

### 6.1 予備実験に関する考察

#### 6.1.1 文節の難易度

予備実験において、文節の表示時間を説明する要因として文節の難易度は不要であるとした。この原因として考えられることは使用した辞書に含まれる単語数が少なかったことである。予備実験で使用された文節の難易度分布を以下の表とグラフに示した。

表 6.1: 予備実験に使用した文節の難易度

難易度	0	1	2	3	4	5	6	7	合計
文節数	82	83	98	193	402	172	38	757	1825

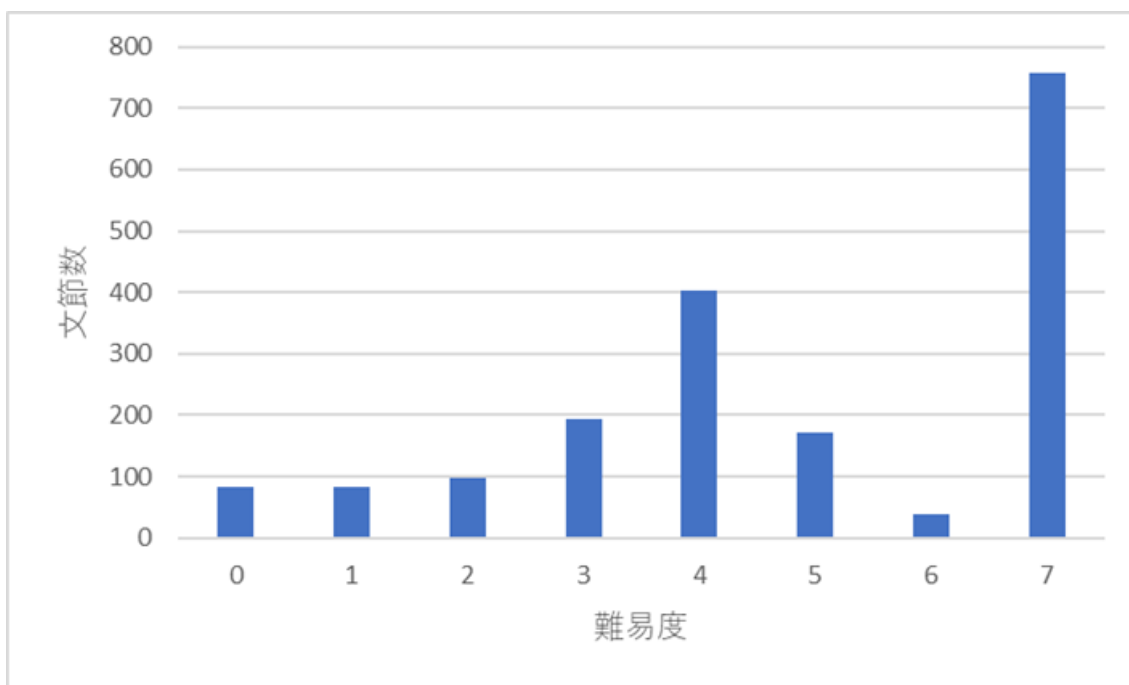


図 6.1: 表示した文節の難易度別分布



この表から難易度7とされた文節の数は全体の約41%であることが分かった。この難易度7に分類された単語をいくつか見てみると理解できないような単語は含まれていないことから、辞書に含まれる単語の数が少なかったと考えられる。

また、文節で難易度を決定したことにも問題があると考えられる。文章が難しいというのは単語が難しいということも原因としてあるが、文章のつながりなども原因として考えられるので、単に文節に含まれる単語の難易度を使用しただけでは有意な結果は得られないと考えられる。

### 6.1.2 漢字の影響

予備実験の章で示した式3.3から日本語の読むときの時間に影響する要素が分かる。文節に含まれる漢字の数の係数は[文節の文字数-文節に含まれる漢字の数]の係数の約3倍であるので、漢字の影響はその他の要素の約3倍といえる。

しかし、表3.2と表3.3を比較すると、文節の表示時間の要因として文節の文字数を用いるモデルより、[文節の文字数-文節に含まれる漢字の数]と[文節に含まれる漢字の数]という2つの要因とするモデルにするほうが補正 $R^2$ が高くなっている被験者は4人で、低くなっている被験者は2人ということが分かる。文節に含まれる漢字の数を要素として考えるとモデルとして当てはまりの悪くなる影響が出た2名の被験者の読み速度は6人の中の上位2名と一致しており、この2名は文節に含まれる漢字の影響をほとんど受けることなく日本語を認識することができ、さらにそれが読み速度の向上の要因となっているのではないかと考えられる。

## 6.2 評価実験に関する考察

### 6.2.1 自動表示の効果

予備実験では被験者がスペースキーを押して次の文節を表示したが、評価実験では自動で文節を表示した。その結果、6.2に示すようにすべての被験者で快適速度が向上している事がわかる。原因としては文節を読んだからスペースキーを押すという動作の負担が快適速度に影響を与えていたと考えられる。

表 6.2: 予備実験と評価実験の読み速度

被験者	予備実験での速度 (文字/分)	評価実験での快適速度 (文字/分)	増加率 (%)
A	989	1200	21.2
B	427	586	36.7
C	790	2640	234
D	434	513	18.2
E	319	385	20.5
F	656	743	13.2

### 6.2.2 快適速度

評価実験では予備実験によって得られた固有のパラメータと用いて自動で文節を表示した結果、固定幅で区切って表示する方法よりも快適な読み速度が向上した人が6人中4名いたことが分かった。これは、意味の区

切りである文節で区切ったこと，文節ごとに最適な表示時間を割り当てたことが原因として考えられる．

一方読み速度が下がった人や変化しなかった人が2名いた．原因としては文節最適方式では文節の文字数が固定ではなく長くなる場合がある，この場合サッカードが起こり，通常の読みと同じような眼球の運動が必要となることが考えられる．固定長方式では表示される文字数は5で固定されているのでサッカードは起こらないと考えられる．したがって，文節で区切って意味の理解を重視するか，固定長で区切ってサッカードをなるべく起こさないことを重視するかという2つの目的にはトレードオフの関係があると考えられる．このトレードオフ関係の対策としては，サッカードが起こると考えられる長い文節の場合だけ文節を更に細かい単位に分割することなどが考えられる．

### 6.3 応用例

この手法の応用例としては，スマートウォッチや電光掲示板などでの文章を表示することが考えられる．スマートウォッチでは使用するユーザーが1人であることがほとんどなので，ユーザーのパラメータを用いて最適な表示時間に調節することができる．電光掲示板などは複数の人が見るので，式3.3のような被験者全員のデータをまとめた回帰式を用いて，切片の値を殆どの人が理解できるような値に調節することでこの提案手法を利用することができる．

## 第7章 結論

### 7.1 まとめ

本研究では日本語の文章を高速逐次視覚提示するときの最適な表示時間を決定する回帰式を求め、それを元に最適な表示時間を決定して、快適速度を調べた。

自分でボタンを押して次の文節を表示する方法と、自動で文節が表示される方法では被験者全員が自動で文節が表示される方法のほうが快適速度が速いことが分かった。原因としては、文節を見てからボタンを押すという動作が快適速度に影響を与えていたと考えられる。

また文節最適方式と固定長方式では読み速度が向上した人や、遅くなった人がおり、長い文節を表示した時に発生するサッカードの影響だと考えられる。この対策としてはサッカードが起こると考えられる文節ではその文節を更に細かい単位に分割して表示することが考えられる。

応用例としては使用者が1人だと考えられるスマートウォッチなどで個人に最適化された表示得度で表示することや、電光掲示板など複数の人が見るものでは式 3.3 のように平均化された回帰式を用いてより多くの人が快適に読める速度で表示することが考えられる。

### 7.2 今後の課題

文節の難易度という指標ではなく文章全体や文の複雑さなどの情報も得られれば、さらに良い表示時間で日本語の文章を表示できると考えられる。またサッカードが起こると考えられる長い文節の場合だけその文節を分割して表示することでさらに快適速度が向上するのかを検証する必要がある。

また、アイトラッキングができる機器を使用してこの手法で実際に視点がどのように動いているのかを分析することでさらに詳しい分析ができると考えられる。

# 謝辞

本研究は、電気通信大学大学院 情報理工学研究科 情報・通信工学科 コンピュータサイエンスコースの寺田研究室において、寺田 実准教授のご指導のもと行われました。

寺田 実准教授には、本研究の方針やアイデア、卒業論文の書き方など様々な部分でご指導いただきました。心から感謝を申し上げます。

また、寺田研究室のメンバーである修士課程2年の阿部 真之さん、鈴木 佑樹さん、平田 吉久さん、本田 裕人さん、修士課程1年の安部 文紀さん、山本 愛美さん、渡邊 裕貴さん、学部4年の岡川 翔子さん、佐々木 透さん、下澤 一輝さん、藤本 明優さん、村松 啓寛さんには実験の協力やアプリケーションの使用感に対する意見をいただきました。心から感謝を申し上げます。

## 参考文献

- [1] Gary S. Rubin, Kathleen Turano: “Reading without saccadic eye movements”. *Vision Research*, 32, 895-905 (1992).
- [2] Aries Arditi: “Elicited sequential presentation for low vision reading”. *Vision Research*, 39, 4412-4418 (1999).
- [3] 小林潤平, 関口隆, 新堀英二, 川嶋稔夫: “文節間改行レイアウトを有する日本語リーダーの読み効率評価”, *人工知能学会全国大会論文集* 28, 1-4 (2014).
- [4] 川崎祐一郎, 中田豊久: “ページ送りと改行が無い電子ブックリーダー・インターフェースの開発”, *情報処理学会 インタラクション 2013* (2013).
- [5] 水口充, 岡本英之: “1行スクロールを基本とした電子書籍ビューア”, *情報処理学会研究報告*, Vol.2014-HCI-157 N (2014).