

卒業論文

Wikipedia 情報収集による
機械学習的要約手法の開発

Machine Learning in Automatic
Summarization for Wikipedia Article

富山県立大学工学部電子・情報工学科

1415015 小野田 成晃

指導教員 西田 泰伸

平成30年2月6日

記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	単語
全単語数 (TF-IDF)	N
単語 (TF-IDF)	t
文 (TF-IDF)	d
全ての単語の出現頻度 (TF-IDF)	$\sum_{s \in d} n_{s,d}$
文中の単語の出現頻度 (TF-IDF)	$n_{t,d}$
単語が出現する文書数 (TF-IDF)	$df(t)$
単語	w
任意の 1 つの文	s
文の位置	i
i 回目の文に w が出現する回数	$frec(w_i)$
全文書数	n
単語の位置の特徴を表す関数	$pos(w_i)$
任意の状態	s
状態の集合	S
任意の行動	a
行動の集合	A
任意の報酬	r
報酬の集合	R
試行回数	t
割引率	γ
学習率	α
単語の集合	W
文字削減率	$D(W)$
W の読みの長さの合計	$R(W)$
W の文字数	$L(W)$

目次

記号一覧	2
第1章 はじめに	1
§ 1.1 本研究の背景	1
§ 1.2 本研究の目的	2
§ 1.3 本論文の概要	3
第2章 文章自動要約	4
§ 2.1 文章自動要約とは	4
§ 2.2 語の重要度に基づく手法	5
§ 2.3 Basic Summarization Model	6
第3章 文章平易化	8
§ 3.1 文章平易化とは	8
§ 3.2 語彙平易化とは	9
§ 3.3 文章平易化のためリソースと手法	11
§ 3.4 単語難易度の判定	14
第4章 言語処理における機械学習	15
§ 4.1 強化学習とは	15
§ 4.2 強化学習のモデル	15
§ 4.3 強化学習の更新則	17
第5章 開発システムの概要	19
§ 5.1 データ取得部	19
§ 5.2 文字列処理部	20
§ 5.3 機械学習部	21
第6章 数値実験の結果と考察	23
第7章 結論	27
謝辞	28

参考文献	29
付録	31
A 自動要約システムの開発環境	31

はじめに

§ 1.1 本研究の背景

WWW の発達により，Web 上の情報の量・種類ともに膨大となっている．Web 上の情報にアクセスするのが容易になった反面，Web 上でコアな情報を検索・収集することがより困難になることが危惧される．この問題を解決するために以下2つの方向から見ていくことにする．

1) 日本語読解支援

現在，Web にある記事が全て，語彙力や理解度がネイティブの大人と比べ未発達な外国人や子どもにも配慮して提示されているとは言い難い状況である．さらに，専門的な記事では，参考サイト¹のように非専門家向けに考慮された例もあるものの，殆どの記事がその分野の前提知識や専門家間で共有しているルール等がない人向けには考慮されていない．そこで，ユーザに適切な情報を提示して，文章の読解支援を行う試みが幾つか散見される．例えば，英語版 Wikipedia の記事を言語学者の Charles が唱えた Basic English という The Basic Dictionary という平易な英単語を収録した辞書に基づき，平易にしたオンライン百科事典である SImple Wikipedia²が公開されている．

2) 情報の動的提示の試み

また，ユーザや情報を公開する場所に合わせて，文字数や内容を動的に変化させる試みも行われている．例えば，Amazon でのユーザにあった商品を提示するレコメンド機能や Google でのユーザの年齢層や地域によって検索結果を変えるパーソナライズ検索等がある．そして，ニュースの見出し電光掲示板など，限られた文字数で本文の内容を要約する技術も以前から必要性を唱えられている [1]．しかし，これらの試みには幾つか問題がある．

1) リソース・手法の不足

まず，前述した試みの多くは英語で行われており，日本語では英語に比べ平易化や要約ともに言語リソース・手法ともに少ない．この点については後述の本論にて詳しく記す．一例としては前述した Simple Wikipedia も英語版しか公開されておらず，日本語版は存在し

¹<http://wa3.i-3-i.info>

²https://simple.wikipedia.org/wiki/Main_Page

ない。また、日本語では Basic English のような平易な語や文章自体の定義が明確に定められておらず、日本語検定の語彙やチャレンジ小学国語辞典等の子ども向け辞書の語釈文を使う等、研究で使うリソース・手法が整備されてるとは言い難い状況である。

2) 多義性の問題

計算機で、ある言葉を解析するとき幾つかの意味が取れる。例えば、「甘い」という言葉には味覚を表す基本義以外にも物事の状況や人間活動等、様々な意味をとれる。また、これらの拡張義の中には辞書に収録されていない用法も含まれる [2]。そのため、計算機上で文章を正確に解析するのは容易ではない。さらに、言語というのは常に拡張されているので、それらの新語に対しても考慮しなければならない

§ 1.2 本研究の目的

以上のような背景を鑑み、文章からユーザや状況に応じた情報を抽出して提示するためには、新たな角度から研究する必要があると考えられる。しかし、従来の要約・平易化分野では報知的・指示的といった目的に応じた学習方法は取られているが、このような試みは少ない。例えば、先の Google の例のように要約システムにユーザの個性を反映することがあげられる。このような枠組みを作ることで、文章平易化や要約の出力結果によるビックデータ解析に発展可能である。言い換えれば、平易化や要約におけるレコメンドの仕組みを作ることが必要と考えられる。

そこで本稿では、日本語版 Wikipedia を対象として記事を自動要約して、ユーザの文章理解度に合わせ、元記事を要約し、さらに、シソーラスを用いて平易化も行う。そしてその生成文に対してのユーザのレスポンスを学習データとして機械学習を行い、最終的にユーザに適した文章を動的に提示する。これにより自然言語処理と機械学習的手法により、要約文の可読性を動的に調整する基盤技術を開発する。

そして、これらの技術・手法によりユーザのアクティビティを観測して、そのデータを元に機械学習を行って適切な文章を生成することを目的とする。またこの研究により、機械学習と自然言語処理における要約文の生成と文章平易化との複合タスクでの可用性を調査・検討していく。

§ 1.3 本論文の概要

本論文は次のように構成される。まず、第2,3,4章では、用いた本研究の中核を担う文章の自動要約と平易化についての技術や手法とそれに付随する言語処理における機械学習について関連研究を踏まえて述べていく。第2章では文章要約の定義から初め、それを計算機で行う主な手法とプロセスを示す。続く第3章でも同様に、文章作成における平易化についての定義と計算機でそれを行う諸手法・プロセスを示す。第4章では、強化学習の数理的な説明と更新則について述べた後、自然言語処理における利用を説明する。

第5章では本研究のシステム全体の構造と前提となる技術・理論について示す。ここでは前章で説明した理論をいかに統合するかを包括的に示す。続く6章ではシステムの数値実験ならびにその考察について提示する。最後に第7章でまとめと今後の課題を述べる。また、本研究で開発したシステムのデモバージョンのダウンロード法並びに実行方法は付録に記載する。

文章自動要約

§ 2.1 文章自動要約とは

一般的にの要約とは，文章のポイントを短くまとめて示すことをいう．また，長文章のポイントを短くまとめて示した文章も要約と呼ばれる．定義としては以下のものが挙げられる [3]．

- (1) の文章のポイントを含んでいること
- (2) 長さの短い文章であること
- (3) 全体が文章としてまとまっていること

テキスト自動要約（以下自動要約と呼称する）とは自然言語処理の領域に類する研究分野である．自動要約は利用目的に応じて指示的要約と報知的要約の二つのタイプに分けられる．

指示的要約 (Indicative summarization)

原文を読者が読む必要があるかどうかを判断させるため材料としての要約文を生成する．

報知的要約 (Informative summarization)

原文の代わりとなる要約文を生成する．例えば，ニュースや新聞の見出しには指示的要約が一般的に使われる対して，Simple Wikipedia や子ども向け新聞等は元の文章の内容を変えずに文字数や語彙のみを状況に応じて変化させる報知的要約が取られる．

さらに、要約の際の手法も大きく以下の2つに細分化される。

抽出的要約 (Extraction-based summarization)

原文の中から重要な文だけ抜き出し決められた文字数もしくは行数に要約する。こちらはTF-IDF(Term Frequency - Inverse Document Frequency) 法などの手法により文中の重要文をスコアリングする。そして、スコアが高い文から文字数制限を超えるまで選択し抽出する方法が一般的である。この手法で得られた要約文は元文から抜き出しているため文法的欠陥が発生し難く、文章中の深層格について考慮する必要がないので、実装が生成的要約に比べ平易という特徴を持つ。TF-IDF 等の各手法については後述する。

生成的要約 (Abstraction-based summarization)

原文の意味を保ちつつ、要約文を提示する場に合わせて異なる文構造の要約文を一から生成する要約手法である。原文の内容を計算機に理解させ、さらにそれと同一の意味を持つ文を生成することから抽出的要約と比較して難解であり、実用には自然言語処理を始め様々な分野の知見が必要である。

§ 2.2 語の重要度に基づく手法

もっともシンプルかつ一般的な手法であり、古典的なものとしてはH.P.Luhnのテキストの重要文抽出法が有名である。この方法では、原文の中で語の頻度を高頻度、中頻度、低頻度と分けて中頻度が文中での重要語と位置付けその語が含まれる文を抽出する抽出法を示した[4]。

重要語の判定ではTF-IDF法が有名であり、これは出現頻度が高い単語ほど重要であるが、複数の文で横断的に使用されている語は重要ではないというルールに基づき語の重要度を決定する。

TF-IDF

この手法は文章中の語の重要度を決定するために用いられ次式のように表される。

$$f(w, d) = \frac{n_{w,d}}{\sum_{s \in d} n_{s,d}} \quad (2.1)$$

$$idf(w) = \log \frac{N}{df(w)} + 1 \quad (2.2)$$

ここで、 w 、 d はそれぞれ単語と文であり、 $n_{w,d}$ は文 d 中の w の出現頻度を、分母は文 d 内のすべての単語の出現頻度の和を表す。そして N が全文数、 $df(w)$ が単語 w が出現する文数を表す。

名の通り文章中の語 (Term) の出現頻度 (Frequency) と語の逆文頻度 (Inverse Document Frequency) つまり、ある語 w が多くの文を横断して出現している場合、その語の重要度は低くなる。そして、これらの TF(2.1) と IDF(2.2) をかけた値が文中の語の重要度を表す。

§ 2.3 Basic Summarization Model

次に抽出法の一つで You Ouyang らが考案した重要語を重み付けするだけではなく、文中の重要語とその出現回数をペアにして重み付けする Basic Summarization Model という手法を説明する [5]。Luhn が用いた方法では重要語が含まれる文を均一に得点付けしていたのに対し、その重要語が文の何回目の登場かも判断基準に含める手法である。モデルは以下の式によって表される。

$$score'(s) = \sum_i [\log freq(w_i) \cdot pos(w_i)] / |s| \quad (2.3)$$

式 (2.3) ではある一文で出現する単語 w について、文の位置 i (何番目の文か) に w が出現する回数 $freq(w_i)$ と $pos(w_i)$ という単語の位置の特徴を表す関数をかけることで文の得点づけを行い、その文中に出現する単語それぞれに得点づけして、その合計点をその文自体の点数としている。

You Ouyang らの論文では、式 (2.3) の $pos(w_i)$ には文章中の最初に登場する単語はより重要であるという the word position hypothesis (以後、位置説と呼ぶ) のもとで、以下の四つの関数 $pos(i)$ を明示的に選択するようにしている。

(1) Direct proportion (DP)

語の最初の出現には 1 を返し、最後の出現には $\frac{1}{n}$ (n は総文数) を返す。つまり、 i のもとで関数 pos は線形に減少していくという位置仮説上では直感的だと考えられる関数である。また、この関数は次式となる。

$$f(i) = \frac{n - i + 1}{n} \quad (2.4)$$

(2) Inverse proportion (IP)

DP に続いて、この関数も語の位置とその重要度を特徴づけることが可能である。DP より一般的な関数であり、次式となる。この関数では、重要度は逆比率であり、それゆえにより早く減少する。つまり、先行した文を他関数より意味づけられる。

$$f(i) = \frac{1}{n} \quad (2.5)$$

(3)Geometric sequence (GP)

語のそれぞれの出現は次式 (2.6) のような和で表せる．この式は $f(i) = 2 * f(i - 1)$ に派生する可能性がある．すなわち，それぞれの語の初めの出現では重要度は半減する．また， i 回目の出現は $f(i) = (1/2)^{i-1}$ として計算できる．

$$f(i) = f(i + 1) + f(i + 2) + \dots + f(n) \quad (2.6)$$

(4)Binary function (BF)

最後は，値が 1 か λ を取る関数であり，最初の出現とそれ以外を区別し，次式のように最初の出現には重要であると重み付けを行う．なお λ は極小の実数である．

$$f(i) = 1, if i = 1; else \lambda, \quad (2.7)$$

以上の 4 つがあり重要度の勾配は $BG > GS > IP > DP$ の順である．また，DUC(The Document Understanding Conference) という自動要約コンペでは BF 関数が最も高い精度を出した．これらの語の出現を元にスコアリングする手法はニュースの見出しなど指示的な要約では効果を発揮できるが，文章自体の代わりとする報知的な要約では，重要文以外で捨てられる対象が文単位であるため，情報が大きく欠落する可能性がある [6]．そのため，報知的要約では山本らが提案した文ごとに連体修飾要素や比喻表現等を解析して，文内で重要でない節を削除する方法が提案されている [7]．

また，商品レビューに対して文章内の商品の性質・意見をそれぞれ属性値・評価値と定義して 2 値分類の機械学習を行い意見情報抽出する手法も取られている [8]．この他にも原文をオントロジや関係データベースを用いて意味関係を理解させ，再構築する生成的方法がある．これは報知的要約でも効果が期待できるが，抽出法に比べて実現方法が複雑である．

文章平易化

§ 3.1 文章平易化とは

文章の平易化 (Text Simplification) とは, Advaith の論文では以下のように定義されている [9].

「狭義では, テキスト平易化とは原文の意味や情報を保持しつつ, テキストの複雑性を減らすプロセスである。」一部翻訳

平易化の実社会における意義としては, 語の習得が完全ではない子どもや外国人向けでも, 文章を平易化することで理解度が高まること, 専門知識が必要な文書を読む場合, その分野の非専門家でも内容をある程度理解できるように語彙を還元すること等が挙げられる.

一般的に平易にする際の換言候補は類義語もしくは国語辞典の語釈文が挙げられる. そして, それを計算機を用いて行うのが本研究の試みであり, 英語圏では第1章でも述べた Simple Wikipedia を始めとした平易な言語資源が存在し, それらを用いることで語彙の平易化を実現する.

さらに Advaith の論文では, 以下の4つの観点を通して平易化するとされている.

1 語彙 (Lexical)

特定の頭字語の使用を削除し, スペルの標準化を行う. 類義語の使用を除外する. 次に, 特定の代名詞や既出の曖昧な語の参照, 'as' のような曖昧な接続や二重否定を削除する. そして, 否定語の制限をして関係代名詞を含むようにさせる. 最後に数字やデータの標準化を行い, 辞書を設定して曖昧語を削除する.

2 構文 (Syntax)

前置・後置修飾語の使用法を明示することを明文化する. 省略語を削除して, 論説調や指示調に修正する. 名詞郡の量を制限する. 'of' のような特定の前置詞を削除する. 曖昧さを減らすために, 前置詞の位置を明示する. 現在分詞の使用を制限する. 受動態を削除して直接的な表現に訂正する. 特定の接続詞を削除して, 句読点を明示する.

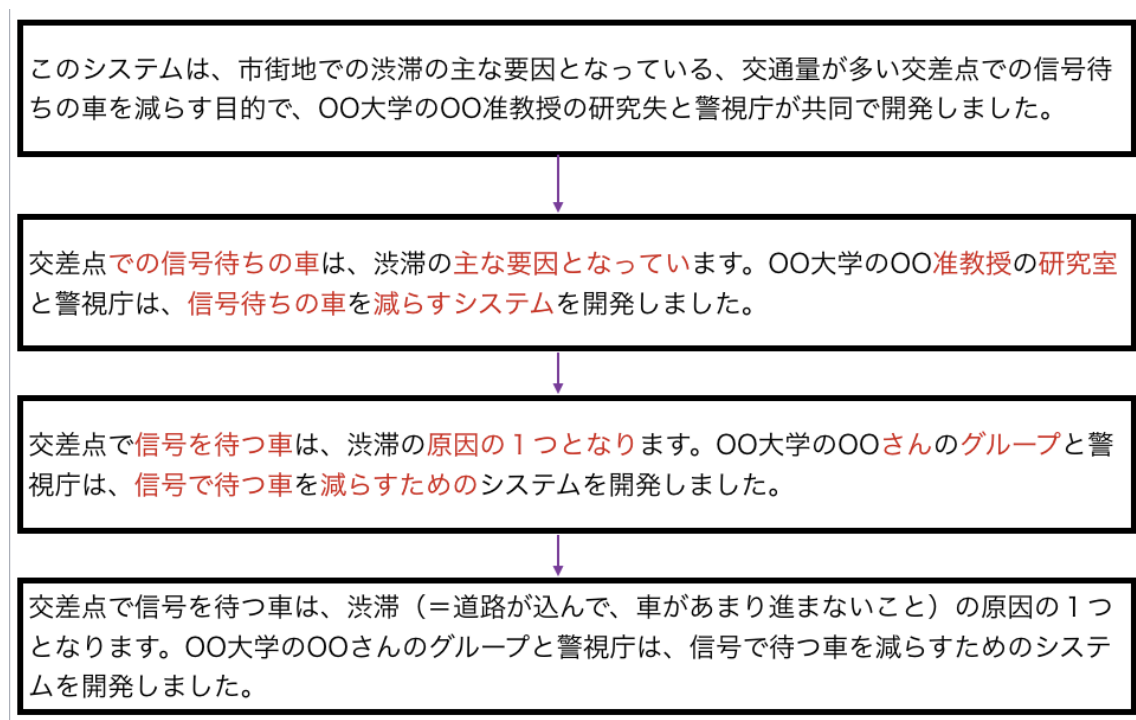


図 3.1 語彙的平易化の変換例

3 文章構造 (Textual Structure)

リストもしくは表を使用した場合は、使用した用語を提示して、文や段落が最長となるようにさせる。また、比喩の使用を制限する。

4 実用 (Pragmatic)

メタファー・スラング・イディオムの使用を避け、可能な限り明確にする。

このルールが多くは日本語でも適応可能であるが、日本語と英語では前置詞の振舞いが異なる等の言語的差異から、日本語で平易化する場合はこれを基本としつつも別の枠組みを作る必要がある。しかし、ルールベースで行うことは言語の複雑な文法や性質を計算機にプログラムする必要もあり、言語資源の少ない言語では、実用は難しいという問題点もある。次に、文章全体ではなく、語彙についての平易化について述べる。

§ 3.2 語彙平易化とは

文章平易化では文章自体の複雑性を減少させる目的で行われていたが、文章中の単語の難解さには着目されていない。そこで、文章中の難解語を平易な語に置き換える語彙平易化 (Lexical Simplification) が必要となる。次に、梶原 [10] の発表から語彙平易化の大まかな手順を説明する。また、平易化手順の例も合わせて図 3.1 に記載する。なお変換例は同発表で使用されたものから抜粋・引用する。

(1) Syntactic Simplification

原文に対して，文圧縮と文分割を行う

(2) Lexical Simplification

語句の言い換えやフレーズベース SMT(Statistical Machine Translation) を実行

(3) Explanation Generation

難易語に対して注釈をつける辞書引きを導入

このようにして原文を要約していく．この方法では分圧縮・文分割はプログラムによる言語処理，二番目の言い換えと三番目の辞書引きは換言語辞書を用意してそれと置き換えることで実現可能である．では次に換言候補を得る具体的な手法について説明していく．梶原らの先行研究から平易化においては語句を置き換えるだけでなく，置き換え語の文章の校正も必要となることが理解出来る．そのため，完全な平易化には 2. の言い換えだけでなく 1,3 のような複数の手法・指標を用いる必要がある．

また，語句の言い換えでは難解な文と平易な文の平行コーパスを用いるのが実用的であるが，日本語の平行コーパスで一般利用可能なものは存在しない．そのため日本語に対しては，美野らのように，語句の難易度の基準について日本語能力試験 (JLPT) の見出し語に対して語釈文を置換単語の候補とする方法や梶原 [11]，山本のように日本語 WordNet 同義語データベースから語彙的還元を行い，述語項構造を用いる方法がある [12]．

しかし，これら手法の問題点として，どの単語が難解か平易かをまず計算機に認識させる必要があり，また，言い換え候補も日本語の場合，例えば「選択」という語が名詞なのか「する」を伴うサ変名詞なのかという細かな文法や語法も計算機が対応しなければならない．さらに，シソーラス辞書のような膨大なデータ資源が必要である．それに対して梶原，小町が新たに考案したコーパスを用いないで単言語平行コーパス構築する手法により，データ資源が少ない言語に対しても，平易化への敷居が下がったと考えられる．

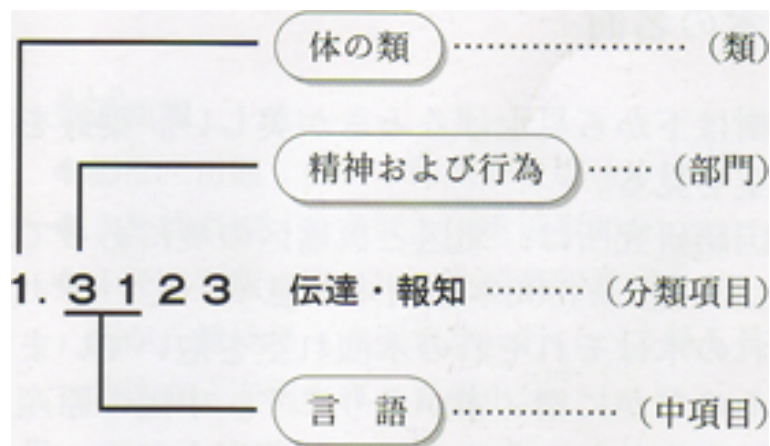


図 3.2 分類番号の構造

2 分類語彙表

日本語分類表³は国立国語研究所資料集により発行された，日本語のシソーラスである．構造としては図 1.3 のように生物の分類に似た構造を取り，「類」「部門」「中項目」「分類項目」からなり，キーワードに対して番号が振られており，そこからその語の分類を把握できる．例によって図 1.3⁴では「情報」という言葉の分類は 1.3123 でこの番号が一致している場合は同義語とみなせる．このように WordNet と同じように同義語を取得できる．しかし同義語と対義語が同一の最小項目に格納されてるため，変換では問題がある．

³http://pj.ninjal.ac.jp/corpus_center/goihyo.html

⁴http://www.ninjal.ac.jp/publication/catalogue/kokken_mado/15/02/

表 3.1: 表 3.1 Word2Vec を用いた類似単語の一例

出力単語	出現確率
春休み	0.8754758834838867
冬休み	0.8532568216323853
休み	0.6718759536743164
修学旅行	0.6634001731872559
宿題	0.6478437781333923
ゴールデンウィーク	0.6392970085144043
遠足	0.6297892928123474
昼休み	0.6146442294120789
休暇	0.6113499402999878
帰省	0.5997079610824585

3 Word2Vec⁵

CBoW(continuous bag-of-word) と skip-gram を元にしたニューラルネットワークで単語ベクトルを学習するモデル。対象の語とその周辺語からニューラルネットワークを構築する。そしてその重み行列、つまり隠れ層を単語のベクトルとして取得する。最後にその隠れ層の単語ベクトルから softmax 関数をかけた出力層に周辺語の出現確率を格納する仕組みとなっている。また、ベクトル化した単語のコサイン類似度を求めることで似ている単語を取得できる。

表 3.1 では著者のコンピュータ上で wikipedia 日本語版の全記事を学習させ、その結果から「夏休み」のコサイン類似度から似ている単語候補を出力した例である。なお実装に関しては web 上の記事⁶を参考にした。しかし、この結果から「冬休み」等、意味としては近いが厳密には類義語とはいえない。そのため、この方法では換言対象の類義語を取得できるとは言い難い状況である。これらの手法で換言に適しているのは 1 の日本語 WordNet 同義語データベースとなる。そこで本研究ではこのデータベースを用いる。

⁵<http://pj.ninjal.ac.jp/corpus.center/goihyo.html>

⁶https://qiita.com/To_Murakami/items/cc225e7c9cd9c0ab641e

表 3.2: 表 3.2 日本語教育語彙表の一部抜粋

No	標準的な表記	読み	語彙の難易度	品詞 1	品詞 2	語種
1448	梅酒	ウメシュ	4. 中級後半	名詞	名詞-普通名詞-	混種語
1449	埋め立て	ウメタテ	4. 中級後半	名詞	名詞-普通名詞-	和語
1450	埋め立てる	ウメタテル	4. 中級後半	動詞 2 類	動詞-一般-	和語
1451	梅干し	ウメボシ	3. 中級前半	名詞	名詞-普通名詞-	和語
1452	埋める	ウメル	4. 中級後半	動詞 2 類	動詞-一般-	和語
1453	羽毛	ウモウ	5. 上級後半	名詞	名詞-普通名詞-	漢語

§ 3.4 単語難易度の判定

平易化の際に単語が難解が平易か判断する必要がある．そこで従来研究として用いられた方法を説明する．

a 国語辞典から換言を得る方法

前述した美濃らの手法 [11] では，ある見出し語の換言候補は国語辞典の語釈文から得られると仮定した研究である．語釈文中の不要語を省くため係り受け解析を行い，パタンを使っている．そして残った主要文の最終文節の自立語を換言候補としている．この方法では，「分析する」というサ変名詞が含まれる文節が「調べるする」という換言になっている．これは「分析」という語の換言候補に「調べる」が挙げられたからである．確かに「分析」と「調べる」の語義は近いが品詞が違うため「する」が含まれるサ変名詞に十分対応できない．

b 日本語教育語彙表

日本語には，Simple Wikipedia のような平易なコーパスは少ない．そこで，語彙に直接レベル付を行う方法がある．それは，李らが開発した日本語教育語彙表 ver1.0⁷を用いる方法である．これは『現代日本語書き言葉均衡コーパス』（BCCWJ:Balanced Corpus of Contemporary Written Japanese）と「日本語教科書コーパス」を元に作成された辞書であり，表 3.2 の通り各単語に 1-6 の難易度レベルと初級後半から上級後半までの 2 つの難易度指標が割り当ててある．これを元に換言候補を決定することができる．

⁷<http://jhlee.sakura.ne.jp/JEL.html>

言語処理における機械学習

§ 4.1 強化学習とは

この章では本研究で用いる機械学習とその周辺知識について述べる。近年、自然言語処理分野でも、コーパスに基づき、統計的手法あるいは機械学習手法を用いて規則なり知識なりを自動獲得し、それを用いた処理を行うことが一般的となっている [15]。本研究でも機械学習を用いるため、その概要と前提となる手法を以下に説明する。

機械学習の中でも、「教師あり学習」と「教師なし学習」、そして「強化学習」に大別される。本研究ではこの中でも強化学習を用いる。強化学習とは、報酬という特別な入力を手がかりに環境に適応した行動決定戦略を追求する機械学習システムと言われている。加えて、強化学習の重要な特徴として以下の2つが挙げられる。

- (1) 1. 報酬駆動形学習であること
- (2) 2. 環境に対する先見的知見を前提としないこと

つまり、学習目標が何であるかをシステムに反映させるだけで、それを実現する方法を獲得させることを意味する。さらに、強化学習では人間の想定以上の解を発見できることがあげられる。例えば、Google 社はビデオゲームの「Atari」を強化学習を組み込んだシステムに実行させることで、ゲームの攻略方法を自動獲得し最終的には人間では実現が難しい程のゲームプレイを見せている¹。このように強化学習は、工学的観点から興味深い枠組みと言われている [16]。次に本研究で用いる具体的な強化学習のモデルについて述べていく。

§ 4.2 強化学習のモデル

木村らの特集記事 [17] を元に強化学習の基本的な性質について述べる。強化学習では環境が用意され、その中での振舞いをエージェントが学習していく、学習方法としては報酬と罰を容易してその値により次に行動すべき行動を学習していく。報酬と罰はスカラー値である。また、報酬には遅延やノイズがある。その為、ある行動を実行直後の報酬だけでは、エージェントはその行動の正しさを判断できない。

¹<https://www.youtube.com/watch?v=V1eYniJ0Rnk>

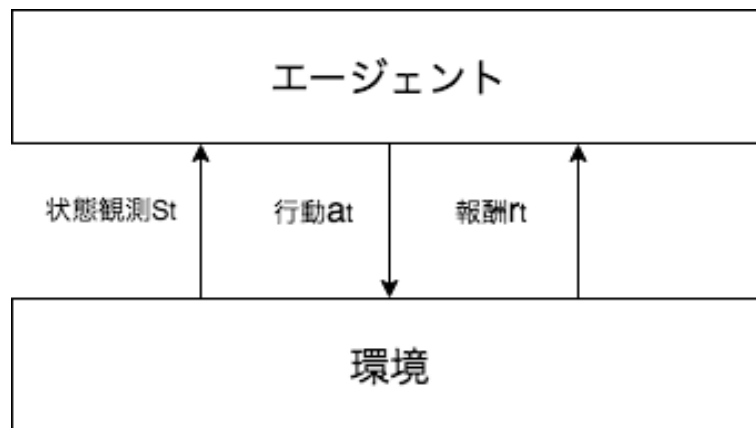


図 4.1 強化学習のモデルのイメージ図

エージェントと環境は図 4.1 のようなやり取りを行う。

エージェントの動作プロセス

1. エージェントは試行回数 t (問題に応じて時間も表すが本研究では施行回数を表す) において環境の状態観測 $S(t)$ に応じて、戦略を組み立て、行動 $a(t)$ を返す
2. エージェントの行動により、環境は $S(t+1)$ へ状態遷移し
その遷移に応じた報酬 $r(t)$ をエージェントに与える。
3. 回数 t を $t+1$ に進めてステップ 1 へ戻る。

エージェントは利得（報酬の総計）の最大化を目的として、状態観測から行動出力の政策 (policy) を獲得する。また、環境とエージェントには下記の性質が期待される。

エージェントの性質

- ・ エージェントは予め環境に対する知識を持たない。
- ・ 環境の状態遷移は確率的。
- ・ 報酬の与えられ方は確率的。
- ・ 状態遷移を繰り返した後、やっと報酬にたどり着くような、段取り的な行動を必要とする環境。

以上が強化学習の基本的な性質の概要である。つぎにこれらの数理的モデルを見ていく。

・ 前提

状態 $s \in S$ 環境がどうなっているか表す観測値

行動 $a \in A$ エージェントが環境に対して起こせる行動

報酬 $r \in R$ 行動を起こした場合に得られる報酬

ただし、この報酬 r は行動した時点でもらえる即時報酬ではなく未来を含めた価値を最大化する報酬とする。そしてある状態 s のもとである行動 a を取ったときの価値を行動価

値観数と呼び次式のようになる。

$$Q(s_t, a_t) = E_{s_{t+1}}(r_{t+1} + \gamma E_{a_{t+1}}(Q(s_t, a_t))) \quad (4.1)$$

*ただし E は期待値を表す

つまり、現時点の行動価値観数 Q は 1 つ先の施行 $t+1$ の Q で示す事ができる。また、ローリスク・ローリターンを取るような行動の偏りを防ぐため、割引率 γ を導入してある。そしてこの Q 値の解の探索方法として挙げられるのが Q -learning, モンテカルロ法, Sarsa 等のアルゴリズムである。

§ 4.3 強化学習の更新則

1 Q -learning

Q -learning(以後 Q 学習とよぶ) は強化学習を、ある試行回数 t で状態 s_t であるとき、取る行動を a_t とした時の行動価値関数を式 (4.1) の期待値の中身 $r_{t+1} + \gamma E_{a_{t+1}}(Q(s_t, a_t))$ を $r_{t+1} + \gamma \max Q(s_{t+1}, a_{t+1})$ と置き換えることで $Q(s_t, a_t)$ で更新するアルゴリズムで次式のように表せる [18]。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \quad (4.2)$$

ここで α は学習率という正のパラメータで一回の学習でどれだけ学習するかを制御する定数。そして、 Q 学習の特徴として次の状態の価値の推定可能な最大値で Q 値を更新する。

2 Sarsa

Q 学習と同様に $TD(Temporal Difference)$ 学習に類するアルゴリズムである。 Q 学習と同様に式 (4.1) の期待値の中身を $r_{t+1} + \gamma \max Q(s_{t+1}, a_{t+1})$ に置き換える。モデルとしては次式のようになる。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \quad (4.3)$$

Q 学習と異なる点として、推定可能な最大値ではなく、次の行動をして、その Q 値を得てから Q 関数を更新する。つまり Q 関数を更新するために 2 回行動が必要となる。

3 モンテカルロ法

上述した 2 つのアルゴリズムと違い、 Q 値の更新時に未来の Q 値を用いず、報酬を得られるまで何度も行動をして、報酬が得られて初めて状態と行動に対して報酬を分配して Q 値を更新する。

Q 関数のアルゴリズムの比較

4.2.3 から 4.2.4 節で確認した手法の特徴をまとめた表を以下表 4.1 にまとめる.

表 4.1: 表 4.1 強化学習の諸手法の比較

	Q 学習	Sarsa	モンテカルロ法
学習速度	×	×	✓
オンライン性 (逐次性)	✓	△	×
ブートストラップ	✓	✓	×
学習の安定性	×	△	✓

ここでブートストラップとは別の状態・行動の価値を使ってある状態・行動の価値を学習することを言う. また, 安定性は Q 学習より方策オン型 (行動方策と評価方策同一であるもの) である Sarsa 法の方が同じ TD 学習でも安定している [19].

開発システムの概要

§ 5.1 データ取得部

本研究ではユーザのアクティティを考慮したシステムの構築を目指している。日本語ではこのような研究は品川らの WWW の自動生成 [20] 等、幾つか存在するがまだ改善の余地があると考えられる。そこで、一から新しいアプローチでシステムを構築していくことにする。本論では、4 章までで研究に必要となる基礎理論や基盤技術をまとめてきた。そして、この章ではそれらの統合して開発する「ユーザのアクティビティを反映した要約システム」について述べる。

本システムでは、対象となる文章の取得・抽出、要約・文章処理、単語レベル設定・シソーラス適用、ユーザによるアクティビティの学習と四つの部に分けられる複合システムである。そこでにおいてそれぞれデータ取得部、要約部、平易化部、機械学習部と位置づける。また、四つの部の実行フローを以下図 5.1 に示す。図 5.1 では機械学習部を実行した後、その結果を要約部以前にフィードバックする。そして以下の節で各部の詳細について示す。

ユーザの文章の理解度や満足度を観測するために以下の 2 つのパラメータを設ける。以後この 2 つのパラメータが再帰的に登場する。

- (1) ユーザレベル (user level): ユーザの語彙力を表す。(1-7 レベルまで)
- (2) 要約率 (compression rate): 加工後の原文に対して何割の要約文を表示するかを表す。(10-100 まで)

この部は変化対象の記事を取得してくる。本研究ではフォーマットの統一のため、対象の記事を Wikipedia の abstract 文に限定した。Wikipedia はクローラーの使用は禁止されているため、直接 Wikipedia の Web サイトから取得するのではなく、Wikipedia から構造化データを抜き出して LinkedOpenData として公開されている DBpedia Japanese¹を使用する。このデータベースは SPARQL という RDF(Resource Description Framework)

¹<http://ja.dbpedia.org>

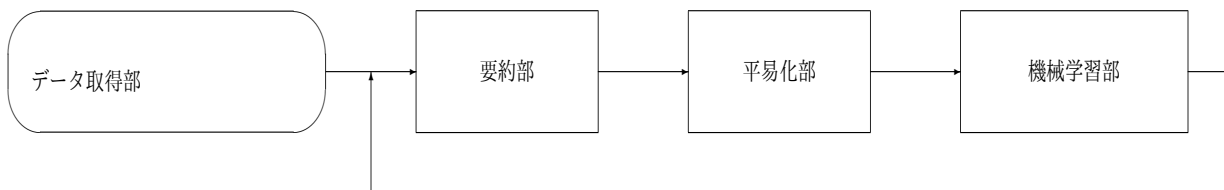


図 5.1: システム全体のフロー図

に対する SQL ライクなクエリ言語を使用して記事を取得する。また、この部で記事中の記号や括弧等の必要ない文字列を削除する。

§ 5.2 文字列処理部

文字列処理部とは、要約部と平易化部からなる。まず、要約部のタスクとして以下を設定した。

- (1) 文章の情報量を動的に変更する
- (2) 文章の可読性（難易度）を動的に変更する

本研究では 2 章でも述べた、You らが考案した Basic Summarization Model[5] を元に抽出法により文章を要約する。これは Luhn が用いた方法では重要語が含まれる文章を均一に得点付けしていたのに対し、その重要語が文の何回目の登場かも判断基準に含める手法であった。このモデルは以下の式によって表される。

$$score'(s) = \sum_i [\log freq(w_i) \cdot pos(w_i)] / |s| \quad (5.1)$$

式 (1) ではある一文で出現する単語 w について、 w の何回目の出現順と $pos(w_i)$ と単語の位置の特徴を表す関数をかけることで文章の得点づけを行い、その文中に出現する単語それぞれに得点づけして、その合計点をその文自体の点数としている。

You らの論文では、式 (5.1) の $pos(w_i)$ には四つの手法 $f(i)$ を明示的に選択するようにしていたが、ここでは一番効果が示された Binary function という手法を用いる。Binary function では w の一番目の出現では $f(i) = 1$ 、それ以外では $f(i) = \lambda$ を返す手法である。なお λ は小さい正の実数で今回は 0.001 とした。

この手法では単語の最初には得点を高くつけ、それ以外では、低い得点を与えることで、新しい単語が出現する文の方が重要性が高いとしている。そのため、Wikipedia のような文頭にその語の定義を述べる百科事典形式のフォーマットとは相性がいいと考えられる。そのためこの手法を採用した。また、日本語の単語抽出する際の形態素解析ツールとして京都大学により開発された MeCab²を使用した。

²<http://taku910.github.io/mecab/>

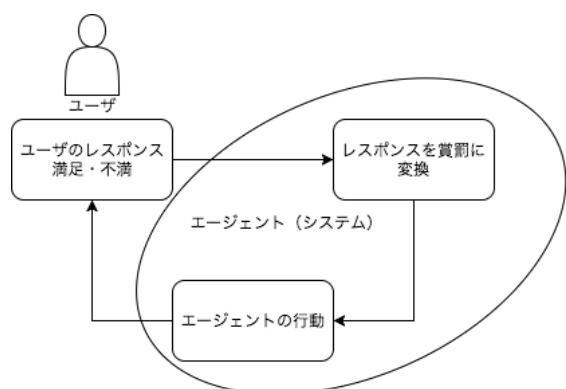


図 5.2: 強化学習の環境とエージェントのイメージ

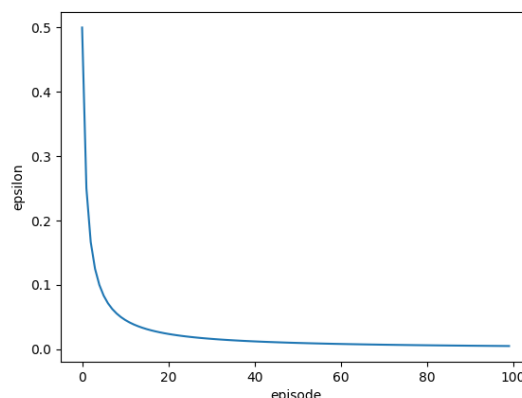


図 5.3: ϵ の値の変化

本研究では第 3 章で述べた手法の内，換言候補が充実している日本語 WordNet を選択することにする．この日本語 WordNet の synsets を同義語と見なし，換言候補に加える．また，本論では要約と平易化の複合タスクのため，文の構成を変化させることは対象外とし，言い換えのみに焦点を当てることにする．具体的な換言方法は以下のプロセスに示す．本システムのプロセス

1. ユーザレベル取得
2. 日本語教育語彙表を用いて，要約文中の語の難易度判定
3. ユーザレベルより高い（難しい）変換候補の単語をマップに格納
4. 類義語の難易度を判定語，ユーザレベルと等しいものを選択
5. 変換候補の単語を平易な類義語で置き換え

なお本稿では，難易度判定に李らの日本語教育語彙表 ver1.0 (<http://jhlee.sakura.ne.jp/JEL.html>) を利用した．またレベル未定義語はレベル 7 に設定した．

§ 5.3 機械学習部

機械学習部ではユーザのレスポンスを逐次反映して学習させるため，強化学習を用いる．本研究における機械学習の環境とエージェントは図 5.2 のようになる．通常の強化学習と異なる点は環境が囲碁や Atari のようなゲームでなく人間（ユーザ）である点である．そのため時間ではなく，試行回数 *episode* を用いて状態遷移させる．ここで，変動するパラメータが要約率 (10-100)，ユーザレベルがあり，アクションはこのパラメータの上昇・減少のため $2 * 2 = 4$ で 4 通り存在する．よって Q 関数のテーブルは $91 * 7 * 4 = 2548$ となり 637 行，4 列の行列となる．

また，関数の探索にはモンテカルロ法と異なり逐次学習でき，さらに Sarsa と異なり方策オフ型なので，取れる戦略の幅が広い Q 学習を採用する．諸値として学習率 $\alpha = 0.5$,

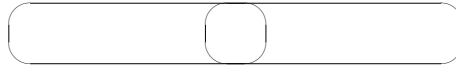


図 5.4 フィードバックのためのスライドバー

割引率 $\gamma = 0.99$ とした．さらに Q 学習の方策として $\epsilon - greedy$ 法を取る．この手法は次式で表せる．

$$\epsilon = \epsilon_0 * (1/episode) \quad (5.2)$$

この ϵ の値によってランダムな行動を取るか，Q 関数から得られた行動を取るか決定する．そして ϵ の値は図 5.3 のような曲線を取るので，施行が増えるにつれランダムな行動は少なくなる．なお本研究では $\epsilon_0 = 0.5$ とした．

ユーザのそのアクティビティに対して実装したのが図 5.4 のスライド式バーである．このバーに付属するボタンを左右に動かすことで文章に対してのフィードバックを得る．このバーのボタンは初期位置はちょうど真ん中に配置されており，ユーザが難解と感じ文章を平易にしたい場合は左に，その逆は右に移動させてフィードバックを直感的に観測できるように設計した．なおバーは左右それぞれ段階まで移動させることで，ユーザの文章に対する不満度を細かくできるようにした．今回は -10 から 10 の移動量を設定した．

これらのフィードバックを機械学習部に組み込む．まず，ユーザの記事に対してのボタンの移動量を取得して，ボタンが満足の移動量が正の値なら報酬を与え，不満の移動量負の値なら罰を与える強化学習的手法を採択した．なお移動量を報酬に変換する場合学習速度を上げるため二乗する．そして，強化学習のフォワード先として，要約率，ユーザレベルの 2 つのパラメータとする．このパラメータの比率を学習により変化させる．

数値実験の結果と考察

構築したシステムに対して数値実験を行った。実験環境としては「強化学習」という探索クエリに対して行う。強化学習システムの諸数値に対しては以下の初期値を割り振った。

要約率: 100%

ユーザレベル: 7レベル（最大レベル）

学習率 α : 0.5

割引率 γ : 0.99

レスポンスに対する重み（報酬）: 報酬²

試行回数 (episode 数): 20 回

今回は試験的な実験なので、著者がユーザとしてシステム内で行動することにした。

今回の実験では、語の難易度を定量的に測る指標として、山村 [21] が提案した以下の定式 (6.1) を用いる。

$$D(W) = \frac{R(W) - L(W)}{L(W)} \quad (6.1)$$

- D : 文字削減率
- W : 単語の集合
- R : W の読みの長さの合計
- L : W の文字数

式 (6.1) では、文章中の読みの長さに着目し、文字数に対して読みが長ければ、情報が圧縮され文章の難易度は高いと評価できる。言い換えると、漢字が多く含まれるほど文章が難しくなるということである。ただし、この指標は定量的に評価できる反面、ユーザの個々の好みを測定できない。そこで、指標を導入する。

次の評価指標としてユーザの満足度の遷移、つまり学習曲線を用いる。これを観測する

ことで、ユーザ個々が提示された要約文に対し、満足しているかどうかを観測できる。以上の評価指標と合わせてユーザの 20 回のアクティビティの観測結果を付す。

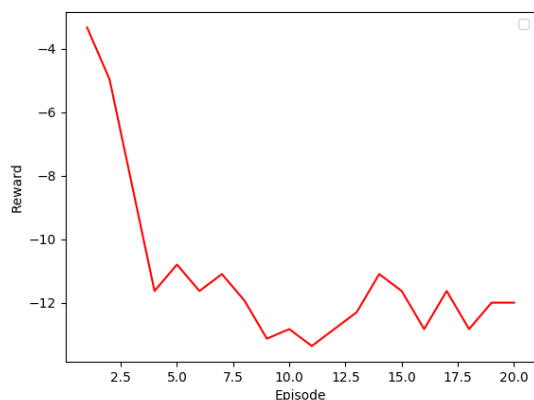


図 6.1: 試行回数 20 回時の学習曲線

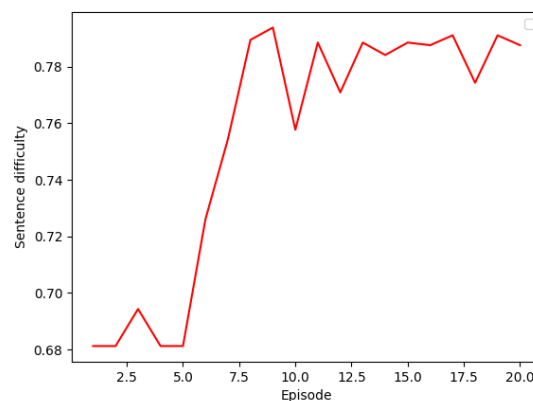


図 6.2: 試行回数 20 回時の文章の難易度の推移

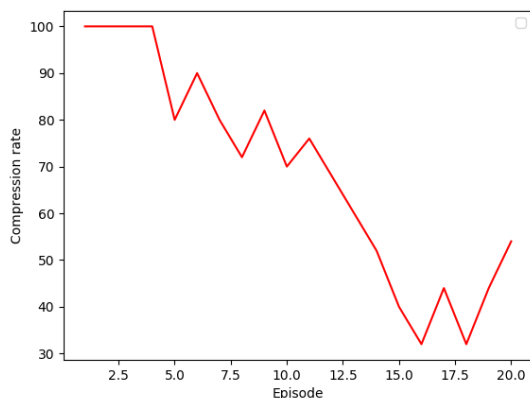


図 6.3: 試行回数 20 回時の要約率の推移

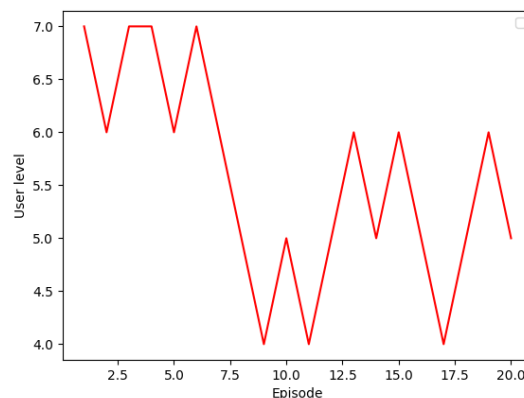


図 6.4: 試行回数 20 回時のユーザレベルの推移

以上の結果から、本研究のシステムと今回の実験について述べる。まず、図 6.1 では学習曲線を表している。学習曲線とは強化学習で得られる各報酬の加算平均の推移を表したものである。これを見ると 2.5 5 回目で大幅に減少してそれから少しずつ上昇と減少を繰り返している。最終的に開始時より低くなっているが、序盤で大きな罰を与えたため、その値に平均が引っ張られている事が考えられる。そのため、今回のシステムの環境設定では賞罰どちらでも大きな値による影響が大きいことが考えられる。これについては環境の初期値を適切な値に設定すれば改善できると思われる。

また、図 6.2 の難易度の推移では 5.0 から 7.5 回にかけて難易度が急上昇しているのが見られる。これは平易化部による変換で音節数を増やしたことが考えられる。例えば、「楽しい」に対して「喜ばしい」と変換した場合、音節数は 4 から 6 に増加する等が挙げられる。これは想定とは異なるので、平易化部の根本からの改善が求められる。図 6.3 では要約率（元の文章の内、何%の文字数があるか）の推移を表している。これを見る限りこのユーザは 100%では不満であり、30%でも不満であることが読み取れる。

最後に、図 6.4 にユーザレベルの推移を示す。このグラフからユーザがどの程度の難易度の語彙を理解できるかを観測できる。全体として、今回の実験では学習曲線が不満（負の値）を示していたが、ユーザの好みをある程度把握できるという点では本システムは有益であると考えられる。

結論

本研究では、ユーザのアクティビティを観測して、文章の難易度や文字数等の特徴を探索するシステムを構築した。また、今回は一連の分野を複合的に組み合わせることに焦点をおいたが、それぞれに対してより細かなチューニングを行うことで、さらなる効果の向上が期待できる。

例えば、要約部としては、Basic Summarization Modelの導入により、スムーズに要約文を提示できた。そして機械学習部では要約率とユーザレベルという2つのパラメータを可変にして文章を動的に生成したが、パラメータを増やすことでユーザのより細かな好みを把握できると考えられる。パラメータの増加につれ計算が複雑になるので、それに合わせてシステムを見直す必要がある。

また、平易化部ではサ変名詞等の日本語の複雑な文法や語法に対応できていなかったもので、形態素解析だけでなく文法ルールを組み込んだルールベースシステムの構築や深層学習での深層生成を用いた文章の全文自動生成も検討される。また、文章の種類は分野に応じ機械学習を用いてのトピック推定を行い、専門用語も適切に換言する必要もある。

以上の通り、本システムの課題は多数存在するが、文章生成時の特徴を動的に変化させること自体の意義は見いだせた。今後さらなる改良を検討していきたい。

謝辞

本研究を遂行するにあたり，多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学電子・情報工学科の西田泰伸准教授に謝意を表します．また，研究に関し，多大なご助力を頂いた奥原浩之教授に感謝します．

最後になりましたが，多大な協力をして頂いた研究室の同輩諸氏に感謝致します．

2018 年 2 月

小野田 成晃

参考文献

- [1] 難波英嗣, 奥村学, “特集 テキスト要約”, 情報処理, vol. 43, NO. 12, pp. 1-8, 2002.
- [2] 姜紅, “コーパスに基づく多義語「甘い」の意味再分類及び語義分布調査”, JCLWorkshop2012_08, pp. 59-68, 2012.
- [3] うちやまかずや “文章の要約”, <<http://web.ydu.edu.tw/~uchiyama/1h93fy/yoyak.html>> アクセス日 2018/02/02.
- [4] Luhn, H. P, “The Automatic Creation of Literature Abstracts”, *IBM Journal of Research and Development*, vol. 2, No. 2, pp. 159-165, 1958.
- [5] You Ouyang, Wenjie Li, Qin Lu, Renxian Zhang, “A Study on Position Information in Document Summarization”, *Colin 2010: Poster Volume*, pp. 919-927, 2010.
- [6] 佐藤理史, 奥村学, “電脳文章要約術 ―計算機はいかにしてテキストを要約するか―”, 情報処理, vol 40, No. 2, pp. 1-5, 1999.
- [7] 山本和英, 増山繁, 内藤照三, “文章内構造を複合的に利用した論説 文要約システム GREEN”, 自然言語処理, vol2, No. 1, pp.1225- 1234, 1995.
- [8] 柏木潔, 小町守, 松本裕治, “レビュー文書からの省略された属性 の推定を含めた意見抽出”, 言語処理学会, 第 19 回年次大会 発表 論文集, pp. 528-531, 2013.
- [9] Siddharthan, A. “A survey of research on text simplification”, *Special issue of International Journal of Applied Linguistics*, 165(2), pp. 1-47, 2014.
- [10] 梶原智之, “語彙的換言を用いたテキスト平易化”, 第七回 NLP 東京 D の会, 2016.
- [11] 美野秀弥, 田中英輝, “国語辞典を使った放送ニュースの名詞の平易化”, 言語処理学会, 第 16 回年次大会 発表論文集, pp. 760-763. 2010.
- [12] 梶原智之, 山本和英, “日本語の語彙平易化システムの構築”, 情報 処理学会, 第 77 回全国大会, pp. 167-178, 2015.
- [13] 梶原智之, 小町守, “平易なコーパスを用いないテキスト平易化の ため単言語パラレルコーパスの構築”, 情報処理学会研究報告 , vol.2016-NL-229, No. 13, pp. 1-8, 2016.
- [14] Xin Rong, “word2vec Parameter Learning Explained”, *Computation and Language*, pp. 1-21, 2016 5.
- [15] 奥村学, “言語処理のための機械学習入門”, コロナ社, 2010
- [16] 宮崎和光, 木村元, 小林重信, “Profit Sharing に基づく強化学習の理論と応用”, 人工知能学会誌, vol.14 No.5, pp. 800-807, 1999.

- [17] 木村元, 宮崎和光, 小林重信, “強化学習システムの設計指針”, 計測と制御, vol.18, No. 10, pp. 1-6, 1999.
- [18] 株式会社ブレインパッド, “強化学習入門 これから強化学習を学びたい人のための基礎知識”, <<http://blog.brainpad.co.jp/entry/2017/02/24/121500>>, アクセス日 2018/02/04.
- [19] いものやま, “強化学習について学んでみた. (その 18)”, <<http://yamaimo.hatenablog.jp/entry/2015/10/15/200000>>, アクセス日, 2018/02/04.
- [20] 品川徳秀, 北川博之, 川田純, “ユーザプロファイルに基づくビューページの動的生成による WWW 閲覧支援”, 情報処理学会論文誌 vol.41 No. SIG6, pp. 22-36, 2000.
- [21] 山村毅, “日本語文章の難易度判定におけるテキスト統計量の有効性”, 電子情報通信学会論文誌 D, Vol. J96D

付録

A 自動要約システムの開発環境

以下に本研究で開発したシステムのデモバージョンのインストールと実行方法を示す。必要なソースコードは (<https://github.com/pomcho555/SwR>) に公開してあるので、各自そちらからダウンロードしていただきたい。

また、必要な環境・リソースは以下がある。

1 Python3.x

Python のインストール方法はいくつかあるが、ここでは、(<https://www.python.org/downloads/>) からバージョン 3.6.4 をインストールすることを推奨する。

2 WordNetJapanese(<http://nlpwww.nict.go.jp/wn-ja/>)

3 goi.csv(<http://jhlee.sakura.ne.jp/JEL.html>)

がある。

なお [2],[3] は二次配布が禁止されているため、デモを動かす場合は各自 URL からダウンロードしていただきたい。

ソースコードと環境が揃った後、ソースと必要なリソースを同じディレクトリ下にまとめる。そして、コンソール上でソースをダウンロードしたディレクトリに移動して、

```
pip install -r requirements.txt
```

これで、必要な Python ライブラリが入手できたので、コンソール上で

```
python rl.py (検索したいワード)
```

で実行ができる。

使い方としては、一度要約文が出力されてその分に対するレスポンスが聞かれるので、-10 から 10 の満足度を入力することにより学習が進行する。

学習をストップしたい場合はレスポンス確認画面で 0000 と入力することによりプログラムが終了して第 6 章で示した各種のグラフが png 形式でカレントディレクトリに保存されるので確認して欲しい。

最後にシステムについて不明な点やバグを発見した場合は github で issue を出すか、t415015@st.pu-toyama.ac.jp まで問い合わせいただけると幸いである。