概要

関連した事柄を調査する際，重要な項目ごとに情報を表に整理することで，その情報を使う人にとって，可読性や利便性が向上すると考えられる．

赤野らの研究 [1] では，word2vec を用いて，複数の文書に出現する単語をベクトルで表現し，これをクラスタリングした後，表の形で整理していた．word2vec では，周辺の単語を考慮してを単語ベクトルを求めるので，周辺に出現する単語の違いによって単語を分類できる．しかし，先行研究では単語のみを表に整理するため，正しく情報を分類できていたとしても，それらの情報がどのような基準で分類されているかが分からず，情報を正確に理解できない場合があった．

本研究では，以上のような問題を解決するために複数の文書から重要な情報を文単位で抽出し表に整理する手法を提案する．提案手法では，文書に含まれる文を意味を崩さない範囲で短い文に分割し，これを単語ベクトルを基にしたベクトルで表現する．そして，得られたベクトルを x-means 法でクラスタリングし，文書ごとに表に整理して表示する．

この提案手法の情報抽出の精度を適合率，再現率，F 値から評価した．また，先行研究によって得られた情報と提案手法によって得られた情報のどちらがより理解しやすい情報であるかを比較した．

提案手法の情報抽出の精度の評価結果は，表から無作為に抽出した 5 列の適合率の平均が 0.91，再現率の平均が 0.64，F 値が 0.73 となった．また，先行手法によって得られた情報と提案手法によって得られた情報のどちらがより理解しやすいかを比較した結果，先行研究に比べ提案手法の方がより情報を正確に理解できるという結果となった．しかし，提案手法の中で行う文の分割の際に行う格解析での解析の誤りによって不自然な文が生成されることがある．重要度の高い上位 5 列に含まれる 147 文のうち 8 文がこのような不自然な文であった．このような不自然な文は理解しづらいため，文の分割方法を見直し，不自然な文が生成されないようにする必要があると考えられる．

目 次

図 目 次

表 目 次

第1章　はじめに

関連した事柄を調査する際，重要な項目ごとに情報を表に整理することで，その情報を使う人にとって，可読性や利便性が向上すると考えられる．

赤野らの研究 [1] では，word2vec を用いて，複数の文書に出現する単語をベクトルで表現し，これをクラスタリングした後，表の形で整理していた．word2vec では，周辺の単語を考慮してを単語ベクトルを求めるので，周辺に出現する単語の違いによって単語を分類できる．しかし，先行研究では単語のみを表に整理するため，正しく情報を分類できていたとしても，それらの情報がどのような基準で分類されているかが分からず，情報を正確に理解できない場合があった．

本研究では，以上のような問題を解決するために複数の文書から重要な情報を文単位で抽出し表に整理する手法を提案する．提案手法では，文書に含まれる文を意味を崩さない範囲で短い文に分割し，これを単語ベクトルを基にしたベクトルで表現する．そして，得られたベクトルを x-means 法でクラスタリングし，文書ごとに表に整理して表示する．

この提案手法の情報抽出の精度を適合率，再現率，F 値から評価した．また，先行研究によって得られた情報と提案手法によって得られた情報のどちらがより理解しやすい情報であるかを比較した．

提案手法の情報抽出の精度の評価結果は，表から無作為に抽出した 5 列の適合率の平均が 0.91，再現率の平均が 0.64，F 値が 0.73 となった．また，先行手法によって得られた情報と提案手法によって得られた情報のどちらがより理解しやすいかを比較した結果，先行研究に比べ提案手法の方がより情報を正確に理解できるという結果となった．

しかし，提案手法の中で行う文の分割の際に行う格解析での解析の誤りによって不自然な文が生成されることがある．重要度の高い上位 5 列に含まれる 147 文のうち 8 文がこのような不自然な文であった．このような不自然な文は理解しづらいため，文の分割方法を見直し，不自然な文が生成されないようにする必要があると考えられる．

本研究の主張点を以下に示す．

新規性

赤野らの研究 [1] などの従来手法では情報を「1 月 20 日」のように単語単位で抽出していたが，本研究では情報を「1 月 20 日に発売する」のように文単位で抽出する．

有用性

文単位の情報を抽出し表に整理することで，単語単位の情報抽出では理解できなかった情報を改善できるという有用性がある．

性能

提案手法の情報抽出の精度の評価結果は，表から無作為に抽出した 5 列の適合率の平均が 0.91，再現率の平均が 0.64，F 値が 0.73 となった．

第2章　関連研究

藤原らの研究 [2] では，複数の文書から重要な情報を CaboCha[1]による固有表現抽出を用いた手法と，ALAGIN[2]の上位下位知識を用いた手法の 2 つの手法により抽出し，抽出した重要情報を表の形で整理していた．また，生成された表の空欄箇所を情報を書き漏らした箇所として検出し，これを指摘し記載の追加を促すことで文章作成支援を行った．

赤野らの研究 [1] では，MeCab[3]によって複数の文書を単語単位に分割し，各単語を word2vec を用いてベクトルで表現した後，これを k-means 法でクラスタリングし，文書を行い，クラスタを列とする表の形で整理していた．また，生成された表の空欄箇所を情報を書き漏らした箇所として検出し，これを指摘し記載の追加を促すことで文章作成支援を行った．

村田らの研究 [3] では，自然言語処理の論文アブストラクトから YamCha と教師あり機械学習を用いて「精度表現」「主要な分野」「言語名」「組織 ・ 人名」の 4 つの重要な表現を抽出した．また，抽出した重要な表現を表に整理したり，重要な表現を含む論文の分布や傾向をグラフで示すことで情報を可視化するツールを構築した．

中渡瀬らの研究 [11] では，論文のアブストラクトの中から「主旨」を表現していると考えられる文を「本研究では」や「本稿では」などのキーワードを基に抽出する．抽出できなかった文に対しては，用意した主旨を表すサンプル文に含まれる述語から作成した述語リストを用いて抽出した．

[1] http://code.google.com/p/cabocha/
[2] http://alaginrc.nict.go.jp/hyponymy/
[3] http://taku910.github.io/mecab/

第3章　提案手法

本研究では図 3.1 のように，関連する内容の複数の文書から情報を抽出し，抽出された情報を文書ごとに表に整理する手法を提案する．

3.1　提案手法の手順

手順 1 複数文書を文単位に分割する．

手順 2 手順 1 で分割された各文の文ベクトルを計算する．

手順 3 文ベクトルを x-means 法 [4, 5] でクラスタリングする．

手順 4 手順 3 で得られた全てのクラスタを見るのは困難なため，クラスタの重要度を計算し，重要度の高い順に，行を文書，列をクラスタとする表に整理することで可読性を高める．

手順 5 表の各列に項目名を付与し，列にどのような情報が含まれるかをわかるようにする．

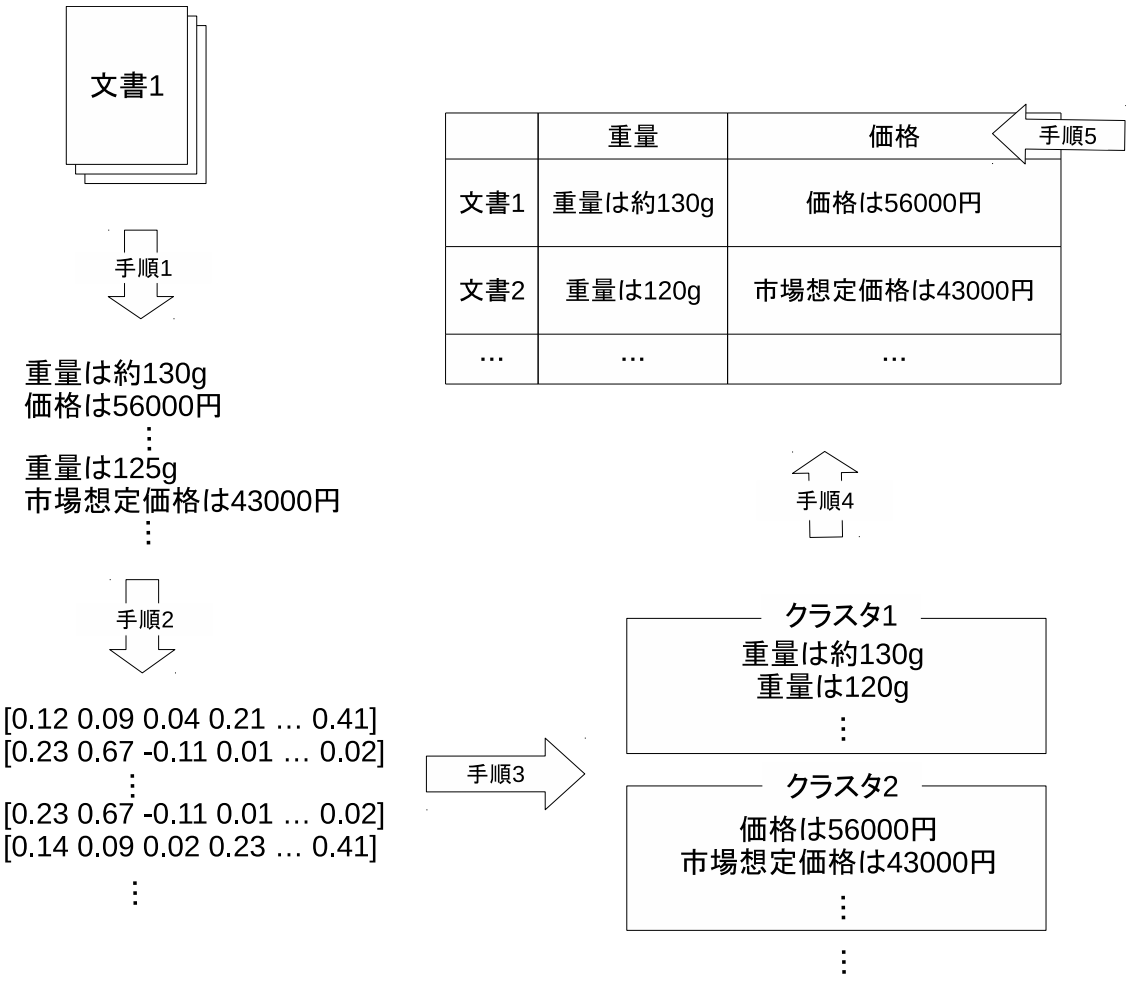

図 3.1: 提案手法の概要図

3.2 文の分割方法

3.1 節の手順 1 における複数文書を文単位に分割する方法を説明する．文書を句点ごとに分割したものを文とした場合，例えば「人口は 98,891 人で，面積は 611.76km^2。」のような複数の情報を含む文が存在してしまう．このような文は,「人口は 98,891 人。」という人口に関する文と,「面積は 611.76km^2。」という面積に関する文に分割されることが望ましい．よって以下の手順で文を分割し，得られた短い文を本研究では 1 つの文として扱う．図 3.2 に分割結果の例を示す．

1. 文を KNP[1]を用いて構文解析する．
2. 条件 (a)，(b) を同時に満たす文節箇所で分割する．
   (a) 文節の係り先が末尾の文節番号である．
   (b) 並列構造を表す<P>が付与されている．
3. 分割された文に対しても，文を分割できなくなるまで 1，2 を行う．
4. 分割された各文を KNP で格解析する．
5. 出力された格解析結果のうち，係り先が末尾の文節番号である文節，もしくは末尾の文節に注目する．
6. 注目している格解析結果に含まれる各格要素について，述語よりも前にある場合は，格要素を格要素に係る文節と統合する．
7. 格要素と述語をまとめて文を作る．

分割前

流域には貴重な生態系が拡がっていたが、噴火によって大半の渓谷が分厚い火山堆積物の底に埋もれた。

分割後

貴重な生態系が流域に拡がっていた
噴火により大半の渓谷が分厚い火山堆積物の底に埋もれた。

図 3.2: 分割結果の例

[1]http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

3.2.1 文ベクトルの計算

3.1 節の手順 2 における文ベクトルの計算方法を説明する．文ベクトルは以下の手順で求める．図 3.3 に文ベクトルの計算手順の例を示す．

1. 文を格要素ごとに分割する．
2. 分割された格要素ごとに以下の手順で格要素ベクトルを求める．
   (a) 文を MeCab[2]を用いて形態素解析する．
   (b) 形態素解析結果のうち，品詞が名詞で，かつ，品詞分類 1 が代名詞，数，非自立，副詞可能でない単語を抽出する．
   (c) 抽出した単語のベクトルの平均を格要素ベクトルとする．
3. 格要素ベクトルの総和を文ベクトルとする．

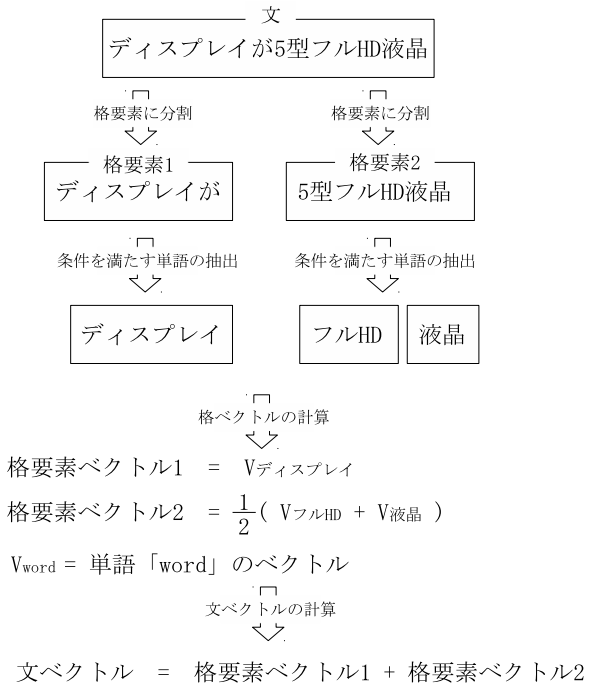

図 3.3: 文ベクトルの計算手順の例

[2]http://taku910.github.io/mecab/

3.2.2 x-means 法

本研究では文ベクトルのクラスタリングに x-means 法を用いる．x-means 法は，k-means 法を拡張した手法である．k-means 法では，あらかじめクラスタの数を指定する必要があるが，x-means 法では以下の手順により，最適なクラスタの数を推測できる．

1. クラスタ数 $k = 2$ で再帰的に k-means 法を実行する．
2. クラスタリング前後のベイズ情報量基準 BIC を比較する．
3. BIC の値が小さくなる限りこれを続ける．

3.2.3 重要度の計算方法

3.1 節の手順 4 におけるクラスタごとの重要度の計算方法を説明する．

クラスタリング結果には表 3.1 の (a) ように関連する文だけで構成される密集率の高いクラスタもあれば，表 3.1 の (b) のように関連性のない文で構成される密集率の低いクラスタもある．密集率の高いクラスタは重要であると考えられる．よって，k 番目のクラスタの密集率 d_k を式 3.1 のように定める．ここで，N_k は k 番目のクラスタに含まれる文の総数であり，$S_{k,l}$ は k 番目のクラスタに含まれる l 番目の文のベクトルであり，$S_{k,mean}$ は k 番目のクラスタに含まれる文のベクトルの平均である．密集率の計算の例を図 3.4 に示す．

$$d_k = \frac{1}{N_k} \sum_{l=1}^{N} \frac{S_{k,l} \cdot S_{k,mean}}{|S_{k,l}||S_{k,mean}|} \tag{3.1}$$

表 3.1: クラスタの密集率の例

(a) 文の密集率が高いクラスタの例

	クラスタ 1
文書1	重量は約130g
文書2	重量は125g
文書3	重量は約140g
文書4	重量は138g

(b) 文の密集率が低いクラスタの例

	クラスタ 2
文書1	重量は約130g
文書2	価格は43000円
文書3	メモリーが3GB
文書4	12月12日に発売する

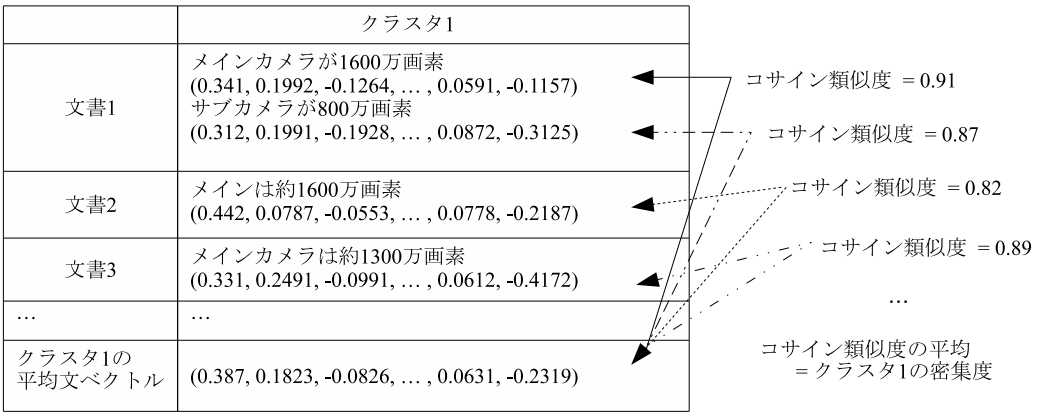

図 3.4: 密集率の計算の例

式 3.1 で求めたクラスタの密集率 d_k を，式 3.2 を用いて，最小値が 0，最大値が 1 になるように正規化する．ここで，nd_k は k 番目のクラスタの正規化されたクラスタの密集率であり，K はクラスタの総数である．

$$nd_k = \frac{d_k - d_{min}}{d_{max} - d_{min}} \tag{3.2}$$

$$d_{min} = \min_{1 \leq k \leq K} d_k \tag{3.3}$$

$$d_{max} = \max_{1 \leq k \leq K} d_k \tag{3.4}$$

多くの文書の情報を含むクラスタほど重要であると考えられる．よって，k 番目の文書カバー率 c_k を式 3.5 のように定める．p_k は k 番目のクラスタにおいて文を抽出できた文書の数であり，P は文書の総数である．

$$c_k = \frac{p_k}{P} \tag{3.5}$$

式 3.5 で求めた文書カバー率 c_k を，式 3.6 を用いて，最小値が 0，最大値が 1 になるように正規化する．ここで，nc_k は k 番目のクラスタの正規化された文書カバー率であり，K はクラスタの総数である

$$nc_k = \frac{c_k - c_{min}}{c_{max} - c_{min}} \tag{3.6}$$

$$c_{min} = \min_{1 \leq k \leq K} c_k \tag{3.7}$$

$$c_{max} = \max_{1 \leq k \leq K} c_k \tag{3.8}$$

k 番目のクラスタの重要度 i_k を式 3.9 のように定義する．

$$i_k = nd_k \times \ nc_k \tag{3.9}$$

3.2.4　クラスタの項目名の求め方

3.1 節の手順 5 におけるクラスタごとの項目名の求め方の概要を図 3.5 に示す．生成された表の各クラスタについて，以下の手順でクラスタの項目名を付与する．

1. クラスタに含まれるの各文について，文に含まれる単語のうち品詞が名詞のものを抽出する．
2. 1 で抽出した各単語について，文書頻度を求める．
3. 文書頻度が最大の単語をクラスタの項目名として付与する．
4. 文書頻度が最大の単語が複数ある場合は，読点で区切って全て付与する．

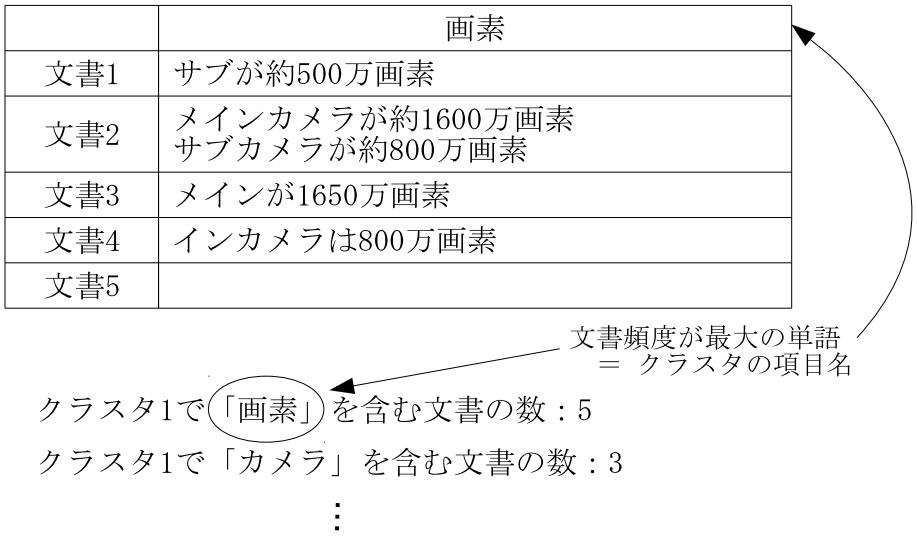

図 3.5: クラスタの項目名の求め方の例

第4章　実験環境

4.1　実験データ

実験では，入力する複数文書データとして，2018 年 1 月 15 日時点での「価格.com」のスマートフォンカテゴリにおける最新の新製品ニュース 30 記事を使用した．文書データの詳細を表 4.1 に示す．また，文書データの例を図 4.1 示す．

表 4.1: 文書データの詳細

文書数	30 文書
文の数	481 文
文字数	23081 文字

UQ コミュニケーションズは、「UQ mobile」に対応する SIM フリースマートフォンのラインアップとして、「HUAWEI nova 2」（ファーウェイ製）を発表。2018 年 1 月下旬以降より、UQ mobile オンラインショップなどで順次、取り扱いを開始する。
ディスプレイに、約 5 型フル HD 液晶（1920 × 1080 ドット）を装備したモデル。メインカメラに、約 1200 万画素と約 800 万画素のダブルレンズカメラを搭載。サブカメラは約 2000 万画素で、同社独自のビューティー補正機能により、立体的かつ正確に顔を認識して、ナチュラルな美肌効果で撮影できる。
このほか、バッテリー容量は 2950mAh で、連続待受時間が約 390 時間、連続通話時間が約 1080 分。プロセッサーは「Kirin659」、メモリーは 4GB。内蔵ストレージは 64GB、外部記録媒体は microSDXC メモリーカード（最大 128GB）をサポート。OS は「Android 7.0」をプリインストールする。
本体サイズは 68.9（幅）× 142（高さ）× 6.9（奥行）mm、重量は約 143g。ボディカラーは、プレステージゴールド、グラファイトブラック、オーロラブルー。

図 4.1: 文書データの例

4.2 MeCab

文の単語への分割には形態素解析器の MeCab を使用した．また，MeCab のシステム辞書には，2017 年 8 月 28 日時点での mecab-ipadic-NEologd[6, 7, 8] を使用した．mecab-ipadic-NEologd では，MeCab の標準のシステム辞書には含まれない固有名詞などの新語を形態素として認識できる.「全国学力テストが行われた」という文を MeCab の標準のシステム辞書と mecab-ipadic-NEologd のそれぞれを用いて分かち書きした結果を以下に示す.

標準のシステム辞書の場合

全国　学力　テスト　が　行わ　れ　た

mecab-ipadic-NEologd の場合

全国学力テスト　が　行わ　れ　た

図 4.2: 辞書による違いの例

4.3 単語ベクトルモデル

3.2.1 節の文ベクトルの計算で用いる単語のベクトルには，fastText[9, 10] によって学習させたものを使用した．fastText は隠れ層と出力層からなる 2 層のニューラルネットワークで，隠れ層が単語の分散表現に相当する.

今回は学習データとして，Wikipedia の全 1,061,375 記事 (2017 年 6 月 1 日時点) を使用した．学習データは前処理としてアルファベットとカタカナは全角に，英数字は半角に統一した．学習データの例を図 4.3 に示す．また，単語ベクトルの次元数は 300 次元とした.

<doc id=”5” url=”https://ja.wikipedia.org/wiki?curid=5” title=”アンパサンド” >

アンパサンド

アンパサンド (, &) とは「…と…」を意味する記号である。英語の に相当するラテン語の の合字で、 (et cetera = and so forth) を と記述することがあるのはそのため。Trebuchet MS フォントでは、と表示され ”et” の合字であることが容易にわかる。

その使用は 1 世紀に遡ることができ (1)、5 世紀中葉 (2,3) から現代 (4-6) に至るまでの変遷がわかる。
Z に続くラテン文字アルファベットの 27 字目とされた時期もある。

アンパサンドと同じ役割を果たす文字に「の et」と呼ばれる、数字の「7」に似た記号があった (, U+204A)。この記号は現在もゲール文字で使われている。

記号名の「アンパサンド」は、ラテン語まじりの英語「& はそれ自身 ”and” を表す」(& per se and) のくずれた形である。英語以外の言語での名称は多様である。

日常的な手書きの場合、欧米でアンパサンドは「 ε 」に縦線を引く単純化されたものが使われることがある。

また同様に、「t」または「+（プラス）」に輪を重ねたような、無声歯茎側面摩擦音を示す発音記号「」のようなものが使われることもある。

プログラミング言語では、C など多数の言語で AND 演算子として用いられる。以下は C の例。
PHP では、変数宣言記号（$）の直前に記述することで、参照渡しを行うことができる。
</doc>

図 4.3: 学習データの例

第5章　実験

5.1　実験結果

4.1 節のデータを入力とした場合に出力された表の上位 6 列を表 5.1 と表 5.2 に示す．出力された表の列の数は 56 個であった．また，表に含まれる分割された文の情報は全部で 620 文あり，これらの文の字数の平均は約 22.5 文字であった．

表 5.1: 出力結果 (1 列目〜3 列目)

	本体サイズ	重量	プリインストール
文書 1	本体サイズは 71 × 144 × 8．5mm、	重量は約 143g	OS は「Android8．0」をする
文書 2	本体サイズは 68．9 × 142 × 6．9mm、	重量は約 143g	OS は「Android7．0」をする
文書 3	本体サイズは 72 × 145 × 8．7mm	重量は約 136g	OS は「Android8．0」をする
文書 4	本体サイズは 10．1mm、	重量は約 150g	OS は「Android7．1」をする
文書 5	本体サイズは 66 × 132 × 9．6mm、	重量は約 140g	OS は「Android8．0」をする
文書 6	本体サイズは 71 × 144 × 8．5mm、	重量は約 143g	OS は「Android8．0」をする
			プリインストールモデル「Google マップ」や「YouTube」といった利用頻度の高いアプリをした
文書 7	本体サイズは 7．6mm、	重量は約 143g	OS は「Android7．0」をする
文書 8	本体サイズは 16．9mm、	重量は約 198g	OS は「Android7．1」をした
文書 9	本体サイズは 10．1mm、	重量は約 150g	OS は「Android7．1」をする
文書 10	本体サイズは 72 × 145 × 8．7mm、	重量は約 136g	OS は「Android8．0」をする
文書 11	本体サイズは 72 × 151 × 12．1mm	重量は約 230g	OS は「Android7．1」をする
文書 12	本体サイズは 50 × 90 × 6．5mm	重量は 38g	
文書 13	本体サイズは 66 × 132 × 9．6mm、	重量は約 140g	OS は「Android8．0」をした
文書 14	本体サイズは 9．7mm	重量は約 140g	OS はプリインストール
文書 15	本体サイズは 75 × 152 × 7．4mm、	重量は 158g	プリインストールバッテリー容量は 3300mAh、OS は「Android8．0」をする

表 5.2: 出力結果 (4 列目〜6 列目)

	メモリー	発売	ディスプレイ
文書 1	メモリーが 3GB、	1 月下旬以降に発売する	
	内蔵ストレージが 32GB		
文書 2	メモリーは 4GB		
文書 3	メモリーが 3GB、	3 月上旬より発売する	
	ストレージが 32GB		
文書 4	メモリーが 3GB、	1 月 19 日より発売する	ディスプレイが 5 型フル HD 液晶
	ストレージが 32GB		
文書 5	メモリーが 3GB	予定だ 1 月下旬より発売する	
	内蔵ストレージが 32GB		
文書 6	メモリーが 3GB、	1 月 18 日に発売する	ディスプレイに、「S3」は光の透過率を高めるとともに消費電力を抑える約 5 型フル HDIGZO 液晶を採用
	内蔵ストレージが 32GB		
文書 7	メモリーが 4GB	1 月下旬より発売する	ディスプレイには、約 5 型フル HD 液晶を装備
文書 8	メモリーが 3GB	3 月下旬より発売する	ディスプレイが約 4．6 型 HD 液晶、
文書 9	メモリーが 3GB、	1 月下旬より発売する	ディスプレイが 5 型フル HD 液晶
	ストレージが 32GB		ディスプレイを割れにくく傷つきにくいガラス DragontrailX にアクリルスクリーンを重ねることで、保護し、
文書 10	メモリーが 3GB	2 月下旬より発売する	ディスプレイが 5．2 型フル HD 液晶× 2、
文書 11	メモリーが 4GB、	2 月下旬より発売する	
	ストレージが 64GB		
文書 12			ディスプレイが「android4．2」をする
文書 13	メモリーが 3GB、	12 月 22 日より発売する	ディスプレイが筐体にフィットする新デザイン「EDGESTfit」も採用した
	内蔵ストレージが 32GB		
文書 14	メモリーが 3GB、	12 月 22 日より発売する	ディスプレイが 4．9 型フル HD ＋液晶、
文書 15	ストレージが 32GB		
	メモリーが 4GB、	12 月 22 日より発売する	ディスプレイが約 6 型有機 EL、
	内蔵ストレージが 128GB		

5.2 情報抽出の評価

式 5.3 より F 値を求め，情報抽出の性能を評価する．式中の「列の項目名を中心とする文」の例を表 5.3 に示す．表 5.3 の 2，3 行目の文は項目名「画素」を中心とする文であるが，4 行目は，項目名「画素」を中心とする文ではないので，列の項目名を中心とする文ではないと判断した．実際に列 20 の適合率と再現率の評価を行った結果を表 5.4 に示す．無作為に抽出した 5 列の評価結果は表 5.5 のようになった．また，重要度の高い上位 5 列の評価結果は表 5.6 のようになった．

$$\text{適合率} = \frac{\text{列の項目名を中心とする文の数}}{\text{列に含まれる文の数}} \tag{5.1}$$

$$\text{再現率} = \frac{\text{列の項目名を中心とする文の数}}{\text{列の項目名を中心とする文書中の文の数}} \tag{5.2}$$

$$F\text{ 値} = \frac{2 \times \text{適合率} \times \text{再現率}}{\text{適合率} + \text{再現率}} \tag{5.3}$$

表 5.3: 列の項目名を中心とした文の例

	画素
文書 1	メインカメラは約 1640 万画素〇
文書 2	カメラ画素数は 500 万画素〇
文書 3	多くの光を集めることでより画質のよい写真を撮影できる ×

表 5.4: 列 20 の適合率・再現率の評価結果

	列 20(実現)	列の項目名を中心とする文	列項目名を中心とする文書中の文
文書 1	3 日間以上の電池稼動を実現	○	3 日間以上の電池稼動を実現
文書 2			立体感のあるサウンドを実現する
文書 3			
文書 4			
文書 5			狭額縁設計により幅 66mm のコンパクトサイズを実現
文書 6	バッテリー容量は 3 日間以上の電池持ちを実現	○	3 日間以上の電池持ち（同社調べ）を実現
文書 7	ファーウェイ独自開発 Histen アルゴリズムによるチューニングで立体感のあるサウンド		
文書 8			
文書 9			
文書 10	耐久性とデザイン性を両立している		
文書 11			
文書 12			
文書 13			
文書 14	挟額縁設計を両立させることで、		実物に近い映像を実現
	モデル片手でも持ちやすいコンパクトサイズを実現した	○	コンパクトサイズを実現
文書 15	表示色域を拡大し、画面占有率 84 %の超狭額縁フルスクリーンを搭載することで、圧倒的な映像美を実現	○	圧倒的な映像美を実現
	高音質を維持しながら、		
	F 値 1．6 のガラスレンズを採用したことで、光の透過率が向上し、		
	ビデオの品質を維持しながら、		
	ダウンロード時間の短縮とストレージの効率化を実現した	○	ダウンロード時間の短縮とストレージの効率化を実現した。
文書 16			
文書 17	約 13 時間のバッテリー駆動を実現している	○	約 13 時間のバッテリー駆動を実現
文書 18			
文書 19			
文書 20			
文書 21			
文書 22	バッテリー容量は 3 日間以上の電池持ちを実現	○	3 日間以上の電池持ちを実現
文書 23	バッテリー容量は 3 日間以上の電池持ちを実現すると	○	3 日間以上の電池持ち（同社調べ）を実現
文書 24	バッテリー容量は 3 日間以上の電池持ちを実現	○	3 日間以上の電池持ちを実現
文書 25			
文書 26			
文書 27			
文書 28	高級感溢れるメタリックな質感を実現	○	高級感溢れるメタリックな質感を実現
文書 29			4K 動画撮影や最高 ISO 感度 12800（静止画）を実現する
文書 30			
計	16	9	14

表 5.5: 無作為に抽出した 5 列の評価結果

列番号	列 2 (重量)	列 9 (バッテリー容量)	列 13 (ボディカラー)	列 20 (実現)	列 34 (円)	平均
適合率	1.00(25/25)	1.00(15/15)	1.00(13/13)	0.56(9/16)	1.00(18/18)	0.91
再現率	0.93(25/27)	0.65(15/23)	0.45(13/29)	0.64(9/14)	0.53(18/34)	0.64
F 値	0.96	0.79	0.62	0.60	0.69	0.73

表 5.6: 上位 5 列の評価結果

列番号	列 1 (本体サイズ)	列 2 (重量)	列 3 (プリインストール)	列 4 (メモリー)	列 5 (発売)	平均
適合率	1.00(25/25)	1.00(25/25)	1.00(27/27)	1.00(44/44)	0.96(24/25)	0.99
再現率	0.96(25/26)	0.93(25/27)	0.87(27/31)	0.60(44/73)	0.71(24/34)	0.81
F 値	0.98	0.96	0.93	0.75	0.81	0.89

5.3　単語単位の情報抽出との比較

単語単位の情報抽出と，文単位の情報抽出で得られた情報がそれぞれ理解できるかを以下の手順で評価する．手順 2 の 4 つの基準の例を表 5.7，表 5.8 に示す．表 5.7，表 5.8 それぞれの列 A はいずれも出力された情報を全て理解できるので◎とする．表 5.7 の列 B は「OS は「android8.0」をする」のような不自然な文が含まれているが，これは OS が何であるかという情報だと推定できる．また，表 5.8 の列 B についても,「スマートフォンが発売される日時」と推定できる．このように，列 B に出力された情報は全て理解できる，もしくは，おおよそ推定できるので○とする．表 5.7，表 5.8 のそれぞれの列 C は文書 1，3，4，5 は情報を理解できるが，文書 2 は情報を理解できないので△とする．表 5.7，表 5.8 のそれぞれの列 D は出力された全ての文が崩れており情報を理解できないので×とする．

重要度の高い上位 20 列での評価結果を表 5.9 に示す．

1. 単語単位・文単位の両方で同じデータを入力として表を生成する．
2. 生成されたそれぞれの表の上位 20 列を以下の 4 つの基準で評価する．

    ◎ 列に出力された全ての情報が理解できる

    ○ 列に出力された全ての情報が理解できる，もしくはおおよそ推定できる

    △ 列に出力された一部の情報が理解できない

    × 列に出力された全ての情報が理解できない

表 5.7: 文単位での評価結果の例

	列 A	列 B	列 C	列 D
文書 1	重量は約 143g	OS は「android8.0」をする	サブカメラが約 500 万画素	機能になる
文書 2	重量は約 143g	OS は「android7.1」をする	背面カメラはなる	画素がある
文書 3	重量は約 136g	OS は「android7.1」をプリインストール	ワイドカメラは約 200 万画素	カメラをする
文書 4	重量は約 150g	OS は「android8.0」をする	サブカメラが約 500 万画素	重量が良い
文書 5	重量は約 140g	OS は「android8.0」をする	メインカメラが約 1650 万画素	OS だ
基準	◎	○	△	×

表 5.8: 単語単位での評価結果の例

	列 A	列 B	列 C	列 D
文書 1	液晶ディスプレイ	1 月 19 日	防水	同社
文書 2	液晶	12 月 14 月	内臓	同社
文書 3	液晶ディスプレイ	1 月 7 日	防水	製
文書 4	液晶	1 月下旬	防水	同社
文書 5	液晶	3 月 15 日	防水	製
基準	◎	○	△	×

表 5.9: 単語単位と文単位の情報抽出の比較

	◎	○	△	×
文単位	0.65(13/20)	0.30(6/20)	0.05(1/20)	0.00(0/20)
単語単位	0.15(3/20)	0.35(7/20)	0.20(4/20)	0.30(6/20)

5.4　項目名の評価

出力された表の列に与えられた項目名が，列の内容を適切に表しているかを評価する．評価には以下の 3 つの基準を用いる．3 つの基準の例を表 5.10 に示す．表 5.8 の 1 列目の項目名「重量」は，列に含まれる文の内容を適切に表しているので，○とする．2 列目に含まれる文の内容から列の項目名は「発売日」が適当だが，項目名「発売」であってもその列に含まれる文の内容をおおよそ推定できるので，列の内容をおおよそ表しているとして△とする．3 列目に含まれる文の内容から項目名は「ボディカラー」が適切だが，与えられた項目名は「用意」となっており，列の内容を表していないので × とする。

重要度の高い上位 20 列を評価した結果を表 5.11 に示す．

○ 列の内容を正確に表している

△ 列の内容をおおよそ表している

× 列の内容を表していない

表 5.10: 項目名の評価結果の例

	重量	発売	用意
文書 1	重量は約 143g	12 月 22 日より発売する	ボディカラーはネイビーを用意
文書 2	重量は約 143g	2 月下旬より発売する	ボディカラーはブラック，ホワイトの 2 色を用意
文書 3	重量は約 136g	3 月上旬より発売する	ボディカラーはブラック，ライトブルー，ホワイトの 3 色を用意
文書 4	重量は約 150g	1 月 19 日より発売する	ボディカラーはライトブルー，ネイビーの 2 色を用意
基準	○	△	×

表 5.11: 項目名の評価結果

○	△	×
0.80(16/20)	0.15(3/20)	0.05(1/20)

第6章　考察

6.1　情報抽出についての考察

5.2 節の表 5.5 から，無作為に抽出した 5 列の適合率の平均は 0.91，再現率の平均は 0.64，F 値の平均は 0.73 であった．また，5.2 節の表 5.6 から，重要度の高い上位 5 列の適合率の平均は 0.99，再現率の平均は 0.81，F 値の平均は 0.89 であった．いずれの場合も適合率に比べ再現率が低い傾向にあった．この原因としては，内容が関連する文が正しく同じクラスタに割り当てられないことが考えられる.「メインカメラは約 1640 万画素」と「約 1650 万画素＋約 1310 万画素のデュアルカメラを搭載」のように,「カメラの画素数」という共通の内容を表す文同士であっても，含まれる単語が大きく異なる場合は，3.2.1 節の方法で文ベクトルを求めると，文ベクトル同士の違いが大きくなる．文ベクトルの違いにより，これらの文が異なるクラスタに割り当てられることで，再現率が低い値となってしまう.

これらの文が同じクラスタに含まれるようにするには，文ベクトルの計算方法を見直する必要がある．提案手法では文中の品詞が名詞で，品詞分類 1 が代名詞，数，非自立，副詞可能でない単語の単語ベクトルを用いて文ベクトルを計算しているが，これらの単語の中には文の持つ情報をよく表す単語もあれば，そうでない単語もある．文の持つ情報をよく表す単語ほど重みを大きくしたうえで，文ベクトルを計算することで，文の情報をより表した文ベクトルが得られると考えられる.

また,「ボディカラーは、オーロラブラックを用意する」という文の「オーロラブラック」という単語は色の種類を表しているが，この文を形態素解析すると「オーロラ」と「ブラック」という 2 つの単語に分かれてしまうため,「オーロラ」という色とは関係ない要素が文ベクトルに含まれてしまう．このような単語を正しく形態素解析するには，より多くの新語に対応したシステム辞書を用いる必要があると考えられる.

6.2　単語単位の情報抽出との比較についての考察

5.3 節の文単位の情報抽出と単語単位の情報抽出との比較では，単語単位の情報抽出よりも文単位の情報抽出のほうが得られた情報を理解しやすいという結果となった．

例えば，単語単位情報抽出では，表 6.1 のように,「13 時間」という情報が得られるが，これが「バッテリーが駆動する時間」についての情報であることが明確でなかった．しかし文単位の情報抽出では表 6.2 のように,「約 13 時間のバッテリー駆動を実現する」のように，得られた情報が「バッテリーが駆動する時間」についての情報であることが明確になり，理解しやすくなった．

しかし，文単位の情報抽出においても「OS は「android8.0」をする」といった文のように理解しづらいものがいくつか見られた．これは正しくは「OS は「android8.0」をプリインストールする」という文で得られるべきであるが， 3.2 節の文の分割で行う格解析での解析の誤りによって不自然な文が生成されている．5.1 節で得られた表では，重要度の高い上位 5 列に含まれる 147 文のうち 8 文がこのような不自然な文であった．

また,「ハイレゾ音源の再生にも対応する」といった文は「（新製品のスマートフォンが）ハイレゾ音源の再生にも対応する」のように，文中の主語が省略されている．このように主語や目的語が省略された文では情報を理解しづらくなる．この問題は，事前に文を照応解析し，省略された主語や目的語を補うことによりある程度解決できると考えられる．

表 6.1: 単語単位の情報の例

単語単位の情報
13 時間
3 日間
3 日間

表 6.2: 文単位の情報の例

文単位の情報
約 13 時間のバッテリー駆動を実現する
3 日間以上の電池持ちを実現
3 日間以上の電池餅を実現

6.3　項目名についての考察

5.4 節の項目名の評価において，△や × と評価された例としては主に，表 6.3 や表 6.4 のようなものがあった．表 6.3 の例では，適切な項目名は「発売日」であるが，そもそも列に「発売日」という単語が含まれないため，適切な項目名が与えられなかった．また表 6.4 の例では，列中で「外部メモリー」と「外部記憶媒体」という同じ意味を持つ二つの単語が統一されることなく使われており，それぞれの単語の頻度を求めるため，項目名としてこれらの単語が与えられなかった．

表 6.3: △と評価した項目名の例 1

発売
12 月 22 日より発売する
2 月下旬より発売する
3 月上旬より発売する
1 月 19 日より発売する

表 6.4: △と評価した項目名の例 2

メモリー，対応
外部メモリーは microSDXC に対応
外部メモリーは microSDXC に対応する
外部記憶媒体は microSD メモリーに対応
外部記憶媒体は microSD メモリーに対応

第7章　今後の課題

5.2 節の表 5.5 から，無作為に抽出した 5 列の適合率の平均は 0.91，再現率の平均は 0.64，F 値の平均は 0.73 であった．また，5.2 節の表 5.6 から，重要度の高い上位 5 列の適合率の平均は 0.99，再現率の平均は 0.81，F 値の平均は 0.89 であった．いずれの場合も適合率に比べ再現率が低い傾向にあったので，再現率の向上を今後の課題としたい．

また，今回の研究では文単位の情報を表に整理することで，単語単位の情報抽出よりも情報が理解しやすくなったが，3.2 節の文の分割で行う格解析での解析の誤りによって生成される不自然な文は理解しずらいものであった．このような不自然な文を無くすために文の分割方法を見直す必要があると考えている．

また，表 7.1 のような字数の多い文の情報を含む列は見づらいため，文の字数を少なくして情報を見やすくする必要があると考えている．

表 7.1: 字数の多い文の情報を含む列の例

	ディスプレイ
文書 1	
文書 2	
文書 3	
文書 4	ディスプレイが 5 型フル HD 液晶
文書 5	
文書 6	ディスプレイに、「S3」は光の透過率を高めるとともに消費電力を抑える約 5 型フル HDIGZO 液晶を採用
文書 7	ディスプレイには、約 5 型フル HD 液晶を装備
文書 8	ディスプレイが約 4．6 型 HD 液晶、
文書 9	ディスプレイが 5 型フル HD 液晶
文書 10	ディスプレイを割れにくく傷つきにくいガラス DragontrailX にアクリルスクリーンを重ねることで、保護し、
文書 11	ディスプレイが 5．2 型フル HD 液晶× 2、
文書 12	ディスプレイが「android4．2」をする
文書 13	ディスプレイが筐体にフィットする新デザイン「EDGESTfit」も採用した

第8章　おわりに

本研究では，先行研究の単語単位の情報抽出では正確に理解できなかった情報を改善するために，関連する複数の文書から重要な情報を文単位で抽出し表に整理する手法を提案した．

提案手法では，文書に含まれる文を意味を崩さない範囲で短い文に分割し，これを単語ベクトルを基にしたベクトルで表現した．そして，得られたベクトルを x-means 法でクラスタリングし，文書ごとに表に整理して表示した．

スマートフォンの新製品ニュース記事 30 点を複数文書とした場合に生成された表の列のうち，無作為に抽出した 5 列の適合率の平均は 0.91，再現率の平均は 0.64，F 値の平均は 0.73 であった．また，重要度の高い上位 5 列の適合率の平均は 0.99，再現率の平均は 0.81，F 値の平均は 0.89 であった．いずれの場合も適合率に比べ再現率が低い傾向にあった．再現率が低くなるのは，文の字面の違いから生じる文ベクトルの違いにより，これらの文が異なるクラスタに割り当てられることが原因と考えられる．

これらの文が同じクラスタに含まれるようにするには，文ベクトルの計算の際に文の情報をよく表す単語ほど重みを大きくしたり，文の構造も考慮することで，文ベクトルをより文の情報を表すようにする必要がある．

また，先行手法によって得られた情報と提案手法によって得られた情報のどちらがより理解しやすいかを比較した結果，先行研究に比べ提案手法の方がより情報を正確に理解できるという結果となった．しかし，提案手法の中で行う文の分割の際に行う格解析での解析の誤りによって不自然な文が生成されることがある．重要度の高い上位 5 列に含まれる 147 文のうち 8 文がこのような不自然な文であった．このような不自然な文は理解しづらいため，文の分割方法を見直し，不自然な文が生成されないようにする必要があると考えられる．

付録 A　文章作成支援

提案手法によって作成した表から重要情報を書き漏らしている文書を見つけ，書き漏らしていることを指摘することで，文書作成を支援することができると考えられる．

例えば，文書 3 に重量に関する情報がない場合，作成された表が表 1 であれば，文書 3 の欄が空欄となっているので正しく書き漏らしを検出できている．一方，表 2 のような表が作成された場合は，文書 3 の欄に重量とは関係のない情報が含まれており，書き漏らしの検出を誤っている．また，表 3 のような表が作成された場合は，本来，重量の情報を含むはずの文書 2 の欄が空欄となっているので，書き漏らしの検出を誤っている．書き漏らし箇所の検出の精度 3 から求める．F 値が大きいほど，書き漏らし箇所を正しく検出できたことを意味する．5.1 節の結果の表の重要度の高い上位 5 列に対する評価結果を表 4 に示す．また，無作為に抽出した 5 列での結果を表 5 に示す．結果から，本来，空欄でない箇所まで空欄となっていることが多かった．この問題を解消するには，6.1 節と同様に，共通の情報を含む文が異なるクラスタに割り当てられることを抑える必要がある．

表 1: 正しく書き漏らしを検出した例

	重量
文書 1	重量は約 130g
文書 2	重量は 125g
文書 3	
文書 4	重量は 138g

表 2: 書き漏らしの検出を誤った例 1

	重量
文書 1	重量は約 130g
文書 2	重量は 125g
文書 3	メモリーが 3GB
文書 4	重量は 138g

表 3: 書き漏らしの検出を誤った例 2

	重量
文書 1	重量は約 130g
文書 2	
文書 3	
文書 4	重量は 138g

$$\text{適合率} = \frac{\text{列の項目名を中心とする文を含まない文書の欄が空欄である数}}{\text{列に含まれる空欄の数}} \tag{1}$$

$$\text{再現率} = \frac{\text{列の項目名を中心とする文を含まない文書の欄が空欄である数}}{\text{列の項目名を中心とする文を含まない文書の数}} \tag{2}$$

$$F\text{ 値} = \frac{2 \times \text{適合率} \times \text{再現率}}{\text{適合率} + \text{再現率}} \tag{3}$$

表 4: 重要度の高い上位 5 列での評価結果

列番号	列 1 (本体サイズ)	列 2 (重量)	列 3 (プリインストール)	列 4 (メモリー)	列 5 (発売)	平均
適合率	0.80(4/5)	0.80(4/5)	0.83(5/6)	0.83(5/6)	0.72(5/7)	0.80
再現率	1.00(4/4)	1.00(4/4)	1.00(5/5)	1.00(5/5)	1.00(5/5)	1.00
F 値	0.89	0.89	0.93	0.93	0.83	0.89

表 5: 無作為に抽出した 5 列での評価結果

列番号	列 2 (重量)	列 9 (バッテリー容量)	列 13 (ボディカラー)	列 20 (実現)	列 34 (円)	平均
適合率	0.80(4/5)	0.48(7/15)	0.11(2/18)	0.89(17/19)	0.84(21/25)	0.62
再現率	1.00(4/4)	0.88(7/ 8)	0.67(2/3)	0.89(17/19)	0.91(21/23)	0.87
F 値	0.89	0.61	0.19	0.89	0.88	0.69

付録B　字数の削減

7章で言及した通り，字数の多い文の情報を含む列は見づらいため，文の字数を少なくして情報を見やすくする必要がある．そこで，以下の手順で文の文字数を少なくした．図1に文字数の削減手順の例を示す．文字数の削減後の例を表6に示す．この方法で字数を削減できた列は全56列中8列のみであった．この方法で字数を削減できるのは，列に含まれる文の構造が非常に似通っている場合のみであるため，一つでも構造が複雑な文を含む列では字数の削減はできなかった．

1. 表の各列を文が含む格要素ごとに分割する
2. 列に含まれる全ての文が同じ格を含む場合は3,4,5の手順を行う
3. 対象となる文の列に含まれるそれぞれの格の前にアルファベットを添えた文字列を全て結合した文字列 (以降, 文パターンと呼ぶ) を求める
4. 格要素の列に含まれる格要素が全て同じである場合はこの格要素の列を消去し,3で求めた文パターンのうち対応するアルファベットを格要素に変換
5. 文パターンを文の列の項目名欄に表示する

表 6: 文字数の削減後の例

	本体サイズは [ガ]X1[述語]	重量は [ガ]X1[述語]
格	述語	述語
文書1	71× 144× 8．5mm、	約143g
文書2	68．9× 142× 6．9mm、	約143g
文書3	72× 145× 8．7mm	約136g
文書4	10．1mm、	約150g
文書5	66× 132× 9．6mm、	約140g
文書6	71× 144× 8．5mm、	約143g
文書7	7．6mm、	約143g
文書8	16．9mm、	約198g
文書9	10．1mm、	約150g
文書10	72× 145× 8．7mm、	約136g

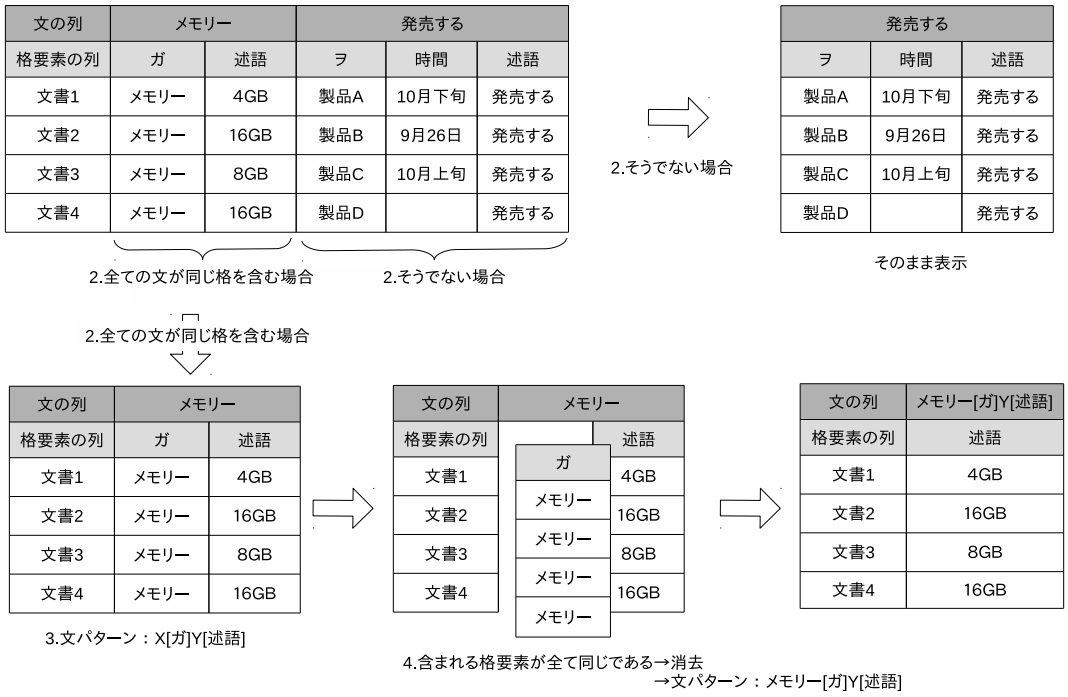

図 1: 文字数の削減手順の例

謝辞

最後に，1 年間に渡りご指導いただきました鳥取大学工学部知能情報工学科自然言語処理研究室の村田真樹教授，村上仁一准教授をはじめ, 自然言語処理研究室の方々に厚く御礼申し上げます．また，参考にさせていただいた論文の著者の方々に対して深く感謝申し上げます.

参考文献

[1] Hokuto Akano, Masaki Murata, and Qing Ma. Detection of inadequate descriptions in wikipedia using information extraction based on word clustering. IFSA-SCIS 2017, pp. 1–6, 2017.

[2] 藤原隆太. Wikipedia からの城情報の取り出しと文章作成支援. Master's thesis, 鳥取大学工学部卒業論文, 2015.

[3] 村田真樹, Stijn De Saeger, 橋本力, 風間淳一, 山田一郎, 黒田航, 馬青, 相澤彰子, 鳥澤健太郎. 論文データからの重要情報の抽出と可視化. 2008-NL-184, pp. 25–32, 2008.

[4] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 727–734, 2000.

[5] 石岡恒憲. x-means 法改良の一提案：k-means 法の逐次繰り返しとクラスターの再併合. 計算機統計学, 第 18 巻, pp. 3–13, 2006.

[6] 奥村学佐藤敏紀. 単語分かち書き辞書 mecab-ipadic-neologd の実装と情報検索における効果的な使用方法の検討. 言語処理学会第 23 回年次大会 (NLP2017), pp. NLP2017–B6–1. 言語処理学会, 2017.

[7] 奥村学佐藤敏紀. 単語分かち書き用辞書生成システム neologd の運用 — 文書分類を例にして—. 自然言語処理研究会研究報告, pp. NL–229–15. 情報処理学会, 2016.

[8] Sato Toshinori. Neologism dictionary based on the language resources on the web for mecab, 2015.

[9] Piotr Bo-janowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. In *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

[10] Armand Joulin, Edouard Grave, Piotr Bo-janowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *arXiv preprint arXiv:1607.01759*, 2016.

[11] 中渡瀬秀一, 大山敬三. 論文アブストラクトからの趣旨抽出方法. 人工知能学会研究会資料 (情報編纂研究会 (第 6 回)), pp. 13–16, 2011.