

# 重要語に着目した深層学習によるレビュー要約方法の提案と評価

2016SE011 原 寛紀 2016SE025 神谷 賢太郎

指導教員 青山 幹雄

## 1 研究背景

近年、EC サイトの商品レビューの増加に伴い、レビュー閲覧の効率化を目的としたレビューの自動要約生成方法が提案されている。しかし、現在広く適用されているクエリ指向要約方法では適切な要約が生成されていないため、内容の効率的な把握が困難である。

## 2 研究課題

前章で述べた現在広く適用されているクエリ指向要約方法では、多くの場合クエリを人手により与えていたため、クエリ作成者の価値観に依存してしまう問題がある。

本研究では、上記の問題を解決するため、以下の3点を研究課題とする。

- (1) レビューからの情報抽出方法の提案
- (2) 抽出した情報を用いたクエリ生成方法の提案
- (3) 自動生成されたクエリを用いた要約の生成と評価

## 3 関連研究

### 3.1 PageRank

Web ページ間のリンク関係を用いて Web ページごとの価値を計算するアルゴリズムである[11]。このアルゴリズムを自然言語である単語や文に適用した TextRank[9]や LexRank[1]は単語や文の重要度の計算が可能である。

### 3.2 RNN(Recurrent Neural Network)

RNN は再帰構造を持つニューラルネットワークである。各時刻の隠れ状態を次の時刻に伝達させることでテキストなどの系列データを扱うことが可能である。

### 3.3 LSTM(Long Short-Term Memory)

LSTMはRNNにおける勾配消失問題を改善するため考案されたニューラルネットワークである。LSTM は RNN の隠れ層を LSTM ブロックに置き換えた構造をしている。

### 3.4 エンコーダデコーダモデル

エンコーダデコーダモデルは入力を固定長のベクトルに圧縮するエンコーダと固定長のベクトルから出力を計算するデコーダから構成される。このモデルは入力と出力の系列が異なる自動要約や自動翻訳の分野で利用される。

### 3.5 アテンション機構

エンコーダデコーダモデルの入力系列に対して入力の情報をより出力に反映させることを可能とする仕組みである[6]。エンコーダの各出力ステップの隠れ状態を保持し、デコーダの各ステップで利用する。

### 3.6 クエリ指向要約モデル

アテンション機構を導入したエンコーダデコーダモデルがある[2, 10]。クエリに対してアテンションを用いることでデコーダの各出力ステップにおいてクエリの情報を出力に反映させることが可能である。本研究では要約にこのクエリ指向要約モデルを使用する。

## 4 アプローチ

本研究では、要約対象の文書から重要語を抽出し、その重要語にアテンションを用いて要約することでデコーダの各出力ステップに重要語の影響を与える要約方法を提案する。文書から抽出した重要語を要約に用いることで重要な内容を漏れなく含む要約の生成を可能とする。重要語の抽出には TextRank アルゴリズムを使用する。重要語からクエリを生成し、クエリ指向要約モデルへ入力することで要約を生成する。アプローチを図1に示す。

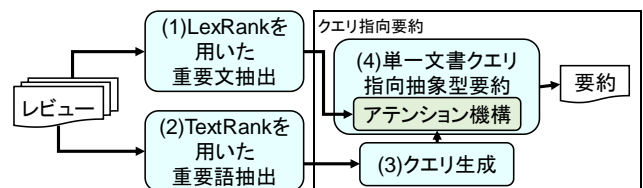


図1 アプローチ

## 5 提案プロセス

### (1) 重要文抽出

重要文抽出プロセスを図2に示す。複数のレビューの各文を TF-IDF を用いてベクトル化する。さらに文同士のコサイン類似度を求める。文をノード、文同士の類似度をエッジとした無向グラフに対して LexRank アルゴリズムを用いて各文の重要度を算出する。重要文抽出プロセスの詳細を以下に示す。

#### (a) 重要文抽出の前処理

TF-IDFにより文同士の類似度を計算するための前処理として形態素解析を行う。

1) ‘.’, ‘!’, ‘?’ など文末で使用される文字を識別し、全てのレビューを文に分割する。

2) 1)で分割した文をさらに単語に分割する。本研究では、英語で記述された文書が対象であり、スペースごとに分割することで各単語が得られる。

#### (b) 文同士の類似度の計算

(a)で分割した文と単語を用いて各単語の TF-IDF の値を算出する。TF-IDF は、文書に含まれる単語の重要度を評価する手法の1つである。レビューの各文を TF-IDF の値を要素としたベクトルで表現し、文同士のコサイン類似度を算出する。

#### (c) 類似度を用いた重要文の抽出

(b)で算出した文同士の類似度を利用して文をノード、エッジを文同士の類似度とした無向グラフを生成する。多くの文と類似し、類似度が高い文を重要な文と定義し、生成した無向グラフから LexRank の値を計算する。LexRank の値が高い順にランキング化し、ランキング上位の文を重要文とする。重要文を抽出しそれらを1つの文書にまとめる。この文書を重

要文書とクエリ指向要約モデルへ入力する。

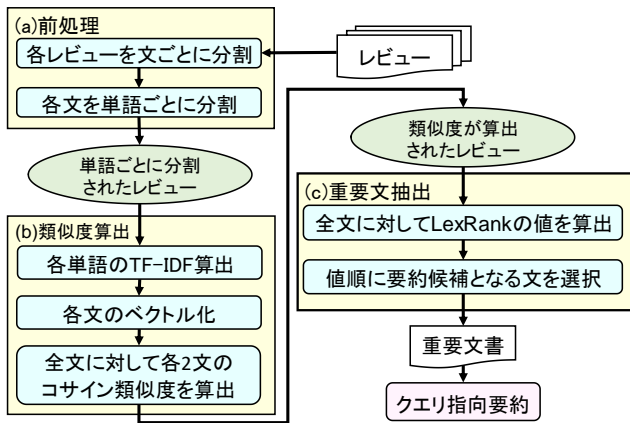


図2 重要文抽出プロセス

(2) 重要語抽出

重要語抽出プロセスを図3に示す。TextRank アルゴリズムを用いてレビューから単語の重要度を算出する。TextRank の値の高い単語を重要語として抽出する。

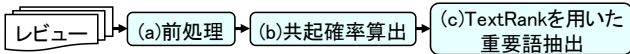


図3 重要語抽出プロセス

(a) 重要語抽出の前処理

レビューから重要語を抽出するための前処理として形態素解析を用いる。複数のレビューを対象とするが、すべてのレビューは同じ商品に対するレビューであるので1つの文書にまとめる。1つの文書にまとめたレビューを形態素に分割する。重要文抽出プロセスにおける形態素解析では、前置詞、接続詞、助動詞、冠詞などの機能語や頻出する単語は排除することなく形態素として分割した。しかし重用語抽出プロセスでは、重要な内容を表す単語を単体で抽出することが目的なので、品詞を一般名詞、固有名詞、動詞、形容詞、副詞に限定し形態素に分割する。

(b) 共起確率算出

(a)で抽出した各単語の共起する頻度を表す共起確率を算出し、共起行列を生成する。すべての品詞に対して共起確率を抽出する場合、'the' や 'a' などの語彙的な意味を含まないが頻出する単語の重要度が高くなってしまふ。共起確率を算出する対象の品詞は(a)の前処理時に限定した品詞とする。

(c) 重要語抽出

(b)で得られた共起確率の情報に基づきノードを単語、エッジを動詞、形容詞、副詞、一般名詞、固有名詞に関する共起確率とした無向グラフを生成する。このグラフを共起グラフとし、TextRank アルゴリズムを共起グラフに適用させて各単語のランク付けを行う。ランキング上位の単語をクエリの候補として抽出する。

(3) クエリ生成

レビューから抽出した重要語の情報をクエリ指向要約モデルへ入力する必要がある。クエリ指向要約モデル

へ入力するクエリは、通常は文書に対する質問文が使われる。提案方法では、抽出した複数の重要語を文字列の形式に変換する。この文字列をクエリとして入力する。

(4) 単一文書クエリ指向抽象型要約

レビュー要約のプロセスを図4に示す。(1)で得られた重要文書を要約対象の文書と(3)で得られた重要語から生成したクエリをクエリ指向要約モデルへの入力とする。エンコーダ、デコーダにはLSTMを用いる。

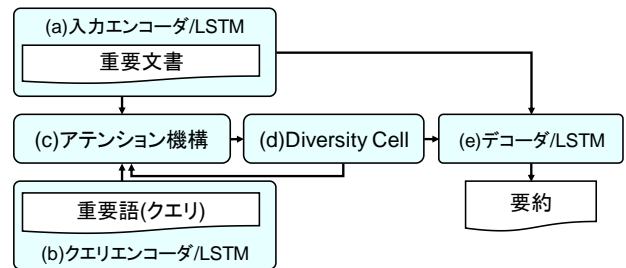


図4 単一文書クエリ指向抽象型要約のプロセス

(a) 入力エンコーダ

重要文抽出プロセスで作成した重要文書を要約対象の文書として入力エンコーダへ入力する。入力エンコーダにはLSTMを使用する。LSTMを用いて実装することで時系列データの長期記憶が可能となり、文脈を考慮した要約を生成できる。エンコーダは入力された重要文書の各単語を先頭から末尾の順に読み込み、全ての単語をベクトル化する。各ステップの隠れ状態は、ベクトル化された単語とひとつ前のステップの隠れ状態から計算する。

(b) クエリエンコーダ

重要語抽出プロセスにおいて生成した重要語で構成されるクエリをクエリエンコーダへ入力する。クエリエンコーダは入力エンコーダと同様にLSTMを使用する。クエリエンコーダがクエリを読み込む過程は入力エンコーダと同様である。

(c) アテンション機構

与えられたクエリに着目させた要約を生成するためにアテンション機構を用いる。デコーダで推測する単語を出力する際、ステップごとに入力された重要文書の着目する箇所を更新する。重要文書とクエリに対してアテンション機構を適用する。

重要文書の各単語に対する重みの状態ベクトルからDiversity Cellに渡す状態ベクトルを計算する。ここで計算されたDiversity Cellに渡す状態ベクトルをコンテキストベクトルと呼ぶ。

(d) Diversity Cell

(c)で計算されたコンテキストベクトルを受け取り、過去のコンテキストベクトルと類似したコンテキストベクトルを渡さないように制御する。これによりデコーダで出力された文が冗長になることを防げる。

(e) デコーダ

重要文書とクエリを用いて要約を推測し出力する。

入力エンコーダ、クエリエンコーダと同様に LSTM を使用する。各ステップの隠れ状態はひとつ前のステップの隠れ状態と出力した単語の情報、コンテキストベクトルから計算する。ソフトマックス関数を用いて出力する単語を推測する。

## 6 プロトタイプの実装

### 6.1 目的

提案方法を評価するために、プロトタイプの実装を行う。プロトタイプの実装を行うことで、従来の要約モデルと提案する要約モデルを比較することが可能となる。

### 6.2 プロトタイプのアーキテクチャ

本研究のプロトタイプのアーキテクチャを図5に示す。

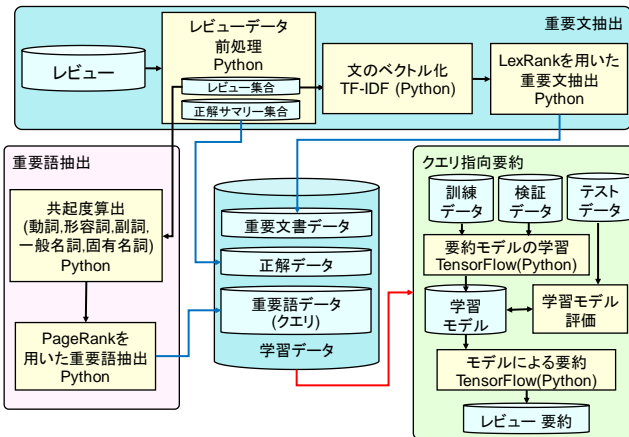


図5 プロトタイプのアーキテクチャ

OSとしてUbuntu, 使用言語としてPython, 深層学習フレームワークとしてTensorFlowを用いて提案アーキテクチャを実装した。

## 7 実データへの適用

### 7.1 適用の目的

プロトタイプの実装後、実際の商品レビューに適用し、提案方法の有効性、妥当性を評価する。

### 7.2 適用対象

Amazonの商品レビューから作成されたデータセットを使用する[3, 7, 8].

使用するデータセットの一例を図6に示す。

```
"reviewerID": "A2SUAM1J3GNN3B",
"asin": "0000013714",
"reviewerName": "J. McDonald",
"helpful": [2, 3],
"reviewText": "I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!",
"overall": 5.0,
"summary": "Heavenly Highway Hymns",
"unixReviewTime": 1252800000,
"reviewTime": "09 13, 2009"
```

図6 使用するレビューの一例

使用データの詳細を表1に示す。

表1 使用データの詳細

asin	商品検索時に使用
reviewText	商品レビュー本文として使用
summary	商品レビュー本文の要約として使用

### 7.3 適用方法

提供されているレビューデータセットは1つの商品に

対するレビュー件数が商品によって様々であるので、レビュー件数が150件程度の商品を対象とした。さらに、要約対象のレビュー数を固定するためにレビューを3件と10件の2パターンに分割した。また、使用データは'reviewText'と'summary'であるので、この2種類のデータを含むデータセットを作成し、プロトタイプを適用した。

### 7.4 適用結果

レビュー件数が3件と10件から重要語をそれぞれ3単語と5単語を抽出した。抽出した重要語と正解の要約内の単語の平均一致率を表2に示す。

表2 正解の要約内の単語と重要語の平均一致率

レビュー件数	抽出した重要語数	平均一致率
3件	3単語	27.9%
10件	5単語	50.6%

レビュー件数が3件と10件から生成した要約文をそれぞれ表3, 表4に示す。表内の<unk>は要約時に使用している単語の辞書に存在しない単語を表す。

表3 レビュー件数3件の要約文

提案方法	i like this one of this product works great with tomato and a <unk> wireless router with a great price
従来方法	phenomenal device for the price great wifi extender great extender

表4 レビュー件数10件の要約文

提案方法	great for the price great little amp great product great for the price great little gadget great for the price great little gadget great for the price great for the price great for the price great sound and great sound great sound great sound
従来方法	kids so so so great but great features, great features, great great great great for great great great great great great for for great great great great great for for for for great great for for for for for great great great great great great great great

プロトタイプの適用によるレビューから要約が生成されるまでの平均単語数の推移を表5に示す。

表5 平均単語数の推移

レビュー件数	(1)レビューの平均単語数	重要文書の平均単語数	(2)要約の平均単語数	要約率 (2)/(1)
3	364.1	87.9	12.1	3.3%
10	1230.6	147.6	45.9	3.7%

## 8 評価

### 8.1 評価の目的と方法

提案方法の有効性、妥当性評価のため、提案方法に対して、クエリを用いない抽象型要約を従来方法として、それぞれの要約モデルで要約を生成した際の Rouge の値[4]を比較することにより評価を行う。学習率は3パターン、隠れ層の次元数も3パターンで要約を生成した。

Rougeとは人手で作成した要約と、システムが生成した要約との一致度の尺度である。本研究では、Rouge-N, とRouge-Lを用いて評価する。Rouge-N, Rouge-Lはどちらも値が高いほど比較する2つの文書の一致度が高いことを表す。

## (1) Rouge-N

人手による要約と推測した要約に対して N-gram 単位で一致度を評価する。Rouge-1, Rouge-2 を用いる。Rouge-N を式[1]に示す。

$$\text{Rouge-N}(C, R) = \frac{\sum_{e \in n\text{-gram}(C)} \text{Count}_{\text{clip}}(e)}{\sum_{e \in n\text{-gram}(R)} \text{Count}(e)} \quad [1]$$

n-gram(C)は、推測した要約に含まれる N グラム、n-gram(R)は、正解の要約に含まれる N グラム集合を表す。Count(e)は、ある N グラムの出現頻度を数える関数であり、Count<sub>clip</sub>(e)は、推測した要約に含まれる N グラムの正解の要約における出現頻度 Count(e ∈ n-gram(C)) と正解の要約における出現頻度 Count(e ∈ n-gram(R)) の小さい方の値を採用する。

## (2) Rouge-L

人手による要約と推測した要約に対して一致する最大のシーケンスを評価する。Rouge-L を式[2]に示す。

$$\text{Rouge-L} = \frac{\sum_{i=1}^m \text{LCS}_U(r_i, C)}{m} \quad [2]$$

m は正解の要約に出現する全単語数、LCS<sub>U</sub>(r<sub>i</sub>, C) は正解の要約中の文 r<sub>i</sub> と推測した要約との間のユニオン LCS スコア[4, 5]を表す。

## 8.2 評価結果

提案モデルと従来モデルの Rouge の値を図7に示す。

Rouge-1, Rouge-2, Rouge-L の全てにおいて従来モデルより提案モデルが高い値を示した。

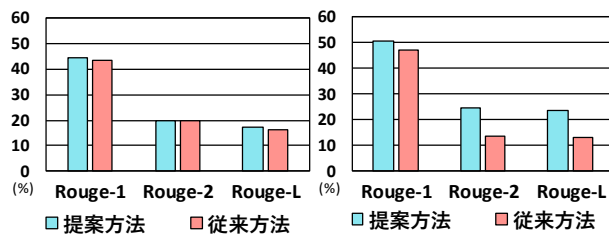


図7 レビュー件数3件(左), レビュー件数10件(右)

## 9 考察

### 9.1 抽出した重要語の妥当性

抽出した重要語は正解の要約との一致率の平均からレビュー件数3件では3割が、レビュー件数10件では5割の重要語が妥当と推定できた。重要語に含まれる単語で正解の要約に含まれなかった単語であっても商品の特徴や評価を表す単語を含むことを目視により確認した。このことから重要語が妥当であると推定できる。

### 9.2 正解の要約と生成した要約との意味の整合性

Rouge-1 の値より、提案方法は従来方法と比べて生成した要約に含まれる単語が正解の要約に含まれる単語と一致度が高いと推定できる。このことから、レビューの内容を容易に把握できると言える。

Rouge-2, Rouge-L の値より、提案方法は要約に含まれる連続する2単語の順序と要約全体の単語の順序を保存していると推定できる。このことから、表現が自然な文章を生成できる。

### 9.3 Rouge から意味の整合性の評価

図7に示した提案方法と従来方法の各 Rouge の値の差はレビュー件数3件の場合より10件の場合が大きい。文書量の増加により各 Rouge の値の差が増大したことから、

提案方法ではクエリに着目することで文書量の増加に関わらず、重要文書の意味の整合性を保存した要約を生成できると推定できる。

### 9.4 生成要約文の内容から意味の整合性の評価

表3, 表4より、レビュー件数が3件の場合、提案方法と従来方法では同一語の反復なく要約を生成できている。10件の場合では、提案方法と従来方法共に同一語の反復が見られる。提案方法では'amp', 'gadget'といったレビュー内で重要な意味を表す語を含んでいる。このことから、提案方法は文書量が増加した場合でも意味の整合性を保つ要約の生成が可能となる。

### 9.5 学習データ量の妥当性

レビュー件数が3件, 10件の場合ともに学習データとして用いた重要文書数は10,000文書を超えている。要約モデルから実際に出力された要約文は、レビュー件数3件の場合に比べ、10件の場合は同一語の反復が多く、かつ、文法的に正しい表現とはなっていない。この原因の1つとして、学習データ量が少ないため、要約モデルの学習が十分でなかったと推定できる。

## 10 今後の課題

今後の課題を以下に示す。

- (1) 重要語の抽出プロセス, 妥当性確認方法の改善
- (2) ハイパーパラメータ最適化による要約精度の向上
- (3) 提案方法の汎用性確認

## 11 まとめ

本研究では、レビューから重要語を抽出し、抽出した重要語を利用した要約の生成方法を提案した。提案方法と従来方法で生成した要約の Rouge の値を比較した結果から、提案方法では意味の整合性が保持された要約を生成可能となる。このことから、レビュー閲覧者は要約の内容を効率的に把握できると期待できる。

## 参考文献

- [1] G. Erkan, et al., LexRank: Graph-based Lexical Centrality as Salience in Text, JAIR, Vol. 22, 2004, pp. 457-479.
- [2] J. Hasselqvist, et al., Query-Based Abstractive Summarization Using Neural Networks, ArXiv, 1712.06100, 2017.
- [3] R. He, et al., Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering, Proc. of WWW 2016, ACM, Apr. 2016, pp. 507-517.
- [4] C.-Y. Lin, ROUGE: A Package for Automatic Evaluation of Summaries, Proc. of Workshop on Text Summarization Branches Out, ACL, Jul. 2004, pp. 74-81.
- [5] C.-Y. Lin, et al., Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics, Proc. of ACL, Jul. 2004, pp. 605-612.
- [6] T. Luong, et al., Effective Approaches to Attention-based Neural Machine Translation, Proc. of EMNLP 2015, ACL, Sep. 2015, pp. 1412-1421.
- [7] J. McAuley, et al., Image-Based Recommendations on Styles and Substitutes, Proc. of SIGIR 2015, ACM, Aug. 2015, pp. 43-52.
- [8] J. McAuley, Repository of Recommender Systems Datasets: Amazon Product Data, <http://jmcauley.ucsd.edu/data/amazon/>.
- [9] R. Mihalcea, et al., TextRank: Bringing Order into Text, Proc. of EMNLP 2004, ACL, Jul. 2004, pp. 404-411.
- [10] P. Nema, et al., Diversity Driven Attention Model for Query-Based Abstractive Summarization. ArXiv, 1704.08300, 2017.
- [11] L. Page, et al., The PageRank Citation Ranking: Bringing Order to the Web, Technical Report, Stanford InfoLab, 1999, pp. 1-17.