文の類似度グラフを用いた複数時系列文書要約

柏井 香里 †　　小林 一郎 ‡
† お茶の水女子大学大学院 人間文化創成科学研究科　‡ お茶の水女子大学基幹研究院
{g1220515, koba}@is.ocha.ac.jp

1 はじめに

ニュースや新聞記事といった時系列文書は次々と新しい情報が追加されていく．そのような文書の全てを読んで理解するには．膨大な時間がかかってしまい現実的ではない．複数の情報源からの文書を要約し，時間の経過とともにその内容を把握できる要約手法が望まれる．本研究ではそのことを踏まえて，複数の新聞社による長期にわたる記事を一つのクロニクルにまとめながら，新しく追加された情報に重きを置いた要約文を時系列順に生成する手法を提案する．

2 時系列文書要約

本研究では，Erkan らによって提案された PageRank[2] に基づいた複数文書要約手法である LexRank[1] を用いる．LexRank は対象文書中の各文をノードとし，ノードをつなぐエッジを文同士の類似性としてグラフを生成する．多くの文と類似している文は重要度が高いという概念のもと，グラフにおける固有ベクトルの中心性の概念に基づいて文の重要度を計算している．Erkan らは，グラフを生成する際に，類似度の値からエッジの重みを利用する重み付きグラフと，閾値を用いて枝刈りを行う重みなしグラフを提案している．本研究では文同士の類似度をエッジの重みとして利用する重み付きグラフを用いる．また，本研究では新しく追加された情報を抽出するため，前日の要約との差分を用いて要約を生成した．提案手法の概要を図 1 に示す．

本研究では文同士の類似度を計算する方法を tf-idf, LDA，word2vec の 3 通り提案する．

2.1 表層情報を用いた複数時系列文書要約

表層情報を用いた要約では，tf-idf を用いた．tf-idf とは文書内の単語に重みを付ける際に用いる手法であり，それぞれの単語の文書内での出現頻度とそれぞれ

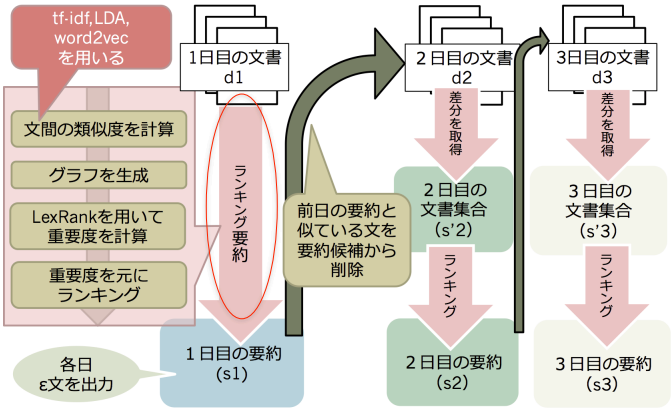

図 1: 提案手法の概要

の単語がいくつの文で使われているかに基づいて計算する．この方法では単語が一致していないと文が類似していると判断できないため，同義語などは判断できない．

2.2 潜在情報を用いた複数時系列文書要約

潜在情報を用いた要約では，LDA(Latent Dirichlet Allocation)[3] を用いた．潜在的ディリクレ配分法 LDA は，Blei ら [3] によって提案された文書中の単語のトピックを確率的に求める手法である．本研究では LDA を応用し，本来なら文書毎に確率を求めているものを文毎にトピックを推定し，文の類似度を測る際に用いている．これによって文中の単語が一致していなくても意味が似ているなら類似していると判断できると考えた．

2.3 word2vec を用いた複数時系列文書要約

文の類似度を計算する際に word2vec[8] を用いた．word2vec はニューラルネットワークを用いて単語の分散表現を獲得する手法であり，Skip-Gram により単語の周辺に現れる単語を予測するモデルによって隠れ

層の重み行列を計算し，その重み行列の各行が単語のベクトルとなる．この単語のベクトルはベクトル同士での演算が可能であり，今回は文のベクトル $v(d)$ を，文中の単語 x の word2vec によるベクトルを $v(x)$ として以下の式 (1) のように計算した．

$$v(d) = \sum_{x \in d} v(x) \tag{1}$$

こちらも意味によって文の類似度を計算できると考えた．

2.4 提案手法のアルゴリズム

LDA や word2vec を用いる際は，文の潜在情報と表層情報どちらも考慮するために，潜在的意味と表層的意味の割合を 0 から 1 までの間の値で変化させる．類似度の計算方法は以下の式 (2) に示す．

$$sim_{score} = \gamma sim_{*} + (1 - \gamma) sim_{surface} \tag{2}$$

sim_{score} は本手法での類似度を意味する．$sim_{surface}$ は ttf-idf を用いた表層情報を意味する．sim_{*} は LDA または word2vec を用いた類似度を意味する．LDA を用いた手法での手順を Algorithm1 に示す．まず，文書集合 $D_t \in \boldsymbol{D}$ について考える．t は時刻単位を表し，t=$\{1,\cdots,T\}$ である．ここで，D_t は時刻 t に属する文書集合を表す．本研究では，時間が経過するとともに新しく文書が追加されることを想定する．入力として，$\boldsymbol{D}$, $\boldsymbol{S}$, ϵ, α を与える．ここで，$\boldsymbol{S}$ は出力する要約の候補となる文集合，α は前日の要約文と当日の文との類似度の閾値であり，ϵ は要約として出力する文の数である．文集合 $\boldsymbol{S}_t$ に含まれる文で構成されるグラフを考える．

また，冗長性の高い要約文の生成を避けるために，MMR [9] を用いる．MMR(Maximal Marginal Relevance) はクエリに基づく文書要約手法として提案された手法であり，クエリの内容との類似性が高く，すでに抽出された文との類似性の低い文を抽出するように定式化されてる，計算式は以下の式 (3) のようになる．

$$MMR' = \underset{s_i \in S \ S'}{\operatorname{argmax}} [score(s_i) - \eta \max_{s_j \in S} sim(s_i, s_j)] \tag{3}$$

Algorithm 1 文の類似度グラフを用いた要約のプロセス

Input: $\boldsymbol{D}$, $\boldsymbol{S}$, ϵ, α, l
$\boldsymbol{S}$ = { }
$\epsilon \leftarrow$ threshold1
$\alpha \leftarrow$ threshold2
for t = 0 to T **do**
  if t=0**then**
    $\boldsymbol{S}_t \leftarrow D_t$
  else
    $\boldsymbol{S}_t$ = { }
    for d to $|D_t|$ **do**
      for s to $|S_{t-1}|$ **do**
        if similarity(d, s) < α **then**
          $\boldsymbol{S}_t \leftarrow d$
        end if
      end for
    end for
    ranking $\boldsymbol{S}_t$ with LexRank based on sentence similarity
    if length of $\boldsymbol{S}_t > \epsilon$ **then**
      $\boldsymbol{S}'_t \leftarrow$ top ϵ sentences of $\boldsymbol{S}_t$
    else
      $\boldsymbol{S}'_t \leftarrow \boldsymbol{S}_t$
    end if
  end if
  $\boldsymbol{S} \leftarrow \boldsymbol{S}'_t$
end for
return $\boldsymbol{S}$

表 1:ニュース資源

トピック	ニュース源	文書数	正解の文数
BP Oil Spill	BBC	293	98
BP Oil Spill	Foxnews	286	52
BP Oil Spill	Guardian	288	307
BP Oil Spill	Reuters	298	30
BP Oil Spill	Washingtonpost	296	19
H1N1 Influenza	BBC	122	40
H1N1 Influenza	Guardian	76	34
H1N1 Influenza	Reuters	207	23
Finiancial Crisis	WP	298	520
Haiti Earthquake	BBC	296	86
Iraq War	Guardian	344	410
Egyptian Protest	CNN	273	55

3 実験

3.1 実験設定

使用したデータ，正解データなど実験に関する設定を記載する．対象データには，Tran ら [4][5] が提供しているタイムライン要約のためのデータセットを用いた．これらは，複数のニュース源から集められた 9 つのトピックに属している新聞記事である．本研究では 9 つのうち 6 つのトピックに関する記事を用いた．表

3 に用いたデータセットの詳細を示す.

また, 前処理として 'a' や 'the' といったストップワードの除去と, ステミング処理を行った. ステミングには Porter の アルゴリズム [6] を用いる. 評価には ROUGE[7] を用い，各新聞社の人手で作成された正解要約をすべて正解データとし，その単語の種類を作成した要約文と比較し単語の一致を見ることで精度と再現率と F 値を計算する．各日毎にそれらの指標とする値を計算し，平均を取ることで全体の要約の性能とした．表 1 に出力文数 ϵ の設定，表 2 に LDA や word2vec を用いる割合 γ 記す．人手で作成された要約を見ると，ほとんどの日の要約文数が 2~4 文であったため，このような設定とした．

表 1: 出力文数の設定

実験 1 ~ 4	
総文数	出力文数
1 ~ 100 文	2 文
101 ~ 500 文	4 文
501 ~ 1000 文	総文数 /100
1000 文以上	10 文

表 2: LDA や word2vec の割合の設定

実験 1	実験 2	実験 3	実験 4
0	LDA 0.5	LDA 0.8	word2vec 0.5

3.2 実験結果と考察

表 3: 実験 1~4 の結果

	再現率	精度	F 値
LexRank	0.72	0.13	0.22
実験 1	0.65	0.29	0.30
実験 2	**0.73**	0.31	**0.38**
実験 3	**0.73**	0.31	0.37
実験 4	0.35	**0.38**	0.28

実験 1~4 の結果は表 3 のようになった．既存の手法である LexRank のみを使った場合と比較して，実験 1~4 はすべて F 値が上回った．表層的意味と潜在的意味の割合を比較すると，実験 1~3 の中では実験 2 が最も F 値が高かったことから，表層的意味と潜在的意味どちらとも使う手法が有効だと分かった．実験 1, 2 と word2vec を用いた実験 4 を比較すると再現率が大きく下がっていた，実際に出力された要約文を見ると，word2vec を用いた手法では単語数の少ない文ばかりが抽出されている事が分かった．単語数が少ない文ばかり抽出したのは，文のベクトルは複数の単語ベクトルの総和になっているので，短い文同士よりも長い文はベクトルが大きく異なっており類似度が低くなりやすいからだと考えられる．よって，文の長さを考慮しないため正確な文の類似をとることができていない．出力された単語数が少ない事で，正解文に含まれる単語を網羅しきれずに再現率が下がったと考えられる．word2vec の性能評価をする場合には出力単語数を同一に設定しないと，正当な評価はできないと考える．このことから，出力文の量を決定する際，文数ではなく単語数によって決めることで再現率を向上させる事が可能だと期待される．

4 まとめと今後の課題

実験結果から，今回提案したグラフと潜在的情報を用いた手法は既存の Erkan らが提案した LexRank よりも性能が良いことが分かった．前日の要約文との差分をとる事で要約の冗長性をなくし，人手で作成する要約に近づける事ができたと分かった．LDA や word2vec などの手法を取り入れる事でさらなる性能の向上がされたが，手法によって出力文数などの設定が最適なものが異なるので，手法毎に最適なものを設定していく事が重要だと分かった．今後の課題として，word2vec を用いた手法での重要文抽出法の改善を目指したい．

参考文献

[1] Gunes Erkan and Dragomir R. Radev, LexRank: Graph-based Lexical Centrality as Salience in Text Summarization, Journal of Artificial Intelligence Research, pp. 457-479, 2003.

[2] Sergey Brin and Lawrence Page, The Anatony of Large-scale Hypertextual Web Search Engine, Computer Networks and ISDN Systems, pp. 107-117, 1998

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan "Latent Dirichlet Allocation", Journal of Machine Learning Research, 3:993-1022, 2003.

[4] G. B. Tran, Tuan A. Tran, N. Tran, M. Alrifai, and N. Kanhabua, Leveraging Learning To Rank in an Optimization Framework for Timeline Summarization, SIGIR, 2013.

[5] G. B. Tran, M. Alrifai, and D. Q. Nguyen, Predicting Relevant News Events for Timeline Summaries, In Proceedings of the 22nd international conference on World Wide Web Companion, pages 91-92.International World Wide Web Conferences Steering Committee, 2013.

[6] M.F. Porter, An algorithm for suffix Stripping, Program, Vol. 14 No.3,pp.130-137, 1980.

[7] C. Lin, ROUGE: a Package for Automatic Evaluation of Summaries, In Proceedings of the Workshop on Text Summarization Branches Out, pp. 74-81, 2004.

[8] Mikolov T., Sutskever I., Chen K., Corrado G. , Dean J, Distributed representations of words and phrases and their compositionality, In Proc. Advances in Neural Information Processing Systems 26 3111-3119 ,2013.

[9] J.Carbonell, Y.Geng, and J.Goldstein. Automated query-relevant summarization and diversity-based reranking. In Proc. of the IJCAI-97 Workshop on AI in Digital Libraries, pp.9-14,1997.