複数時系列文書の単語の出現頻度に基づく重要度とグラフを用いた要約手法への取り組み

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　時系列複数文書の要約

本研究では，上述した時系列文書要約とグラフを用いた手法 A と，単語の特徴量を用いた手法 B の 2 種類の時系列複数文書要約手法を提案する．提案手法の概要を図 1 に示す．

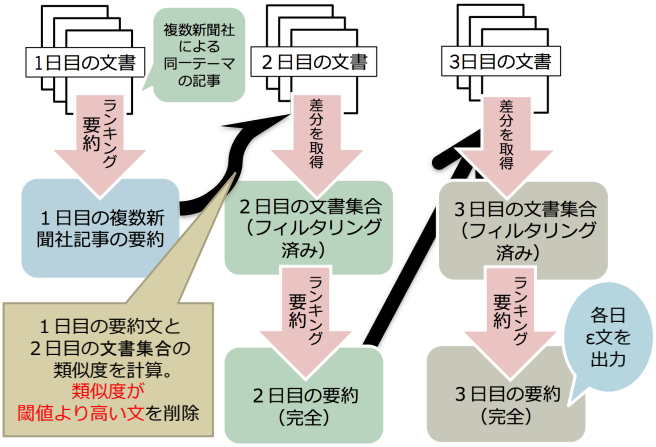

図 1: 提案手法の概要

図 1 には 3 日目までの要約の流れを示してある．複数の新聞社による記事を入力とし，各日毎の要約文を出力する．

2.1　グラフと潜在的情報を用いた要約手法

この手法では，各文の重要度を決定するためにグラフ構造を用いた LexRank という手法を用いる．LexRank は，Erkan ら [1] 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り，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が，本研究では閾値による枝刈りを行わない重みなしグラフを適用する．また，文の類似度を決定する際に，単語の表層的一致と潜在的意味の一致を考え，潜在的意味の抽出には Latent Dirichlet Allocation(LDA)[3] を用いる．

LDA を使用する際にはこれを応用し，本来なら文書単位で確率を求めているものを文単位でトピックを推定し，文の潜在的意味の類似度を測る事を可能にしている．また，文の潜在的意味と表層的意味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．本手法での手順を Algorithm1(手法 A) 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．

Algorithm 1 手法 A の要約のプロセス

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) $< \alpha$ **then**
          $\boldsymbol{S}_t \leftarrow d$
        end if
      end for
    end for
    ranking $\boldsymbol{S}_t$ with LexRank and topicRank
    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 $\boldsymbol{S}$

Algorithm 2 手法 B の要約のプロセス

Input: $\boldsymbol{D}$, $\boldsymbol{S}$, ϵ, α, n, l
$\boldsymbol{S} = \{\ \}$
$top_w = \{\ \}$
$\epsilon \leftarrow$ 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) $< \alpha$ **then**
          $\boldsymbol{S}_t \leftarrow d$
        end if
      end for
    end for
    $top_w \leftarrow$ top n word by tf-idf $\boldsymbol{S}_t$
    for s to $|S_t|$ **do**
      if top_w in s
        $\boldsymbol{score}_s$ += $\boldsymbol{score}_{topw}$
      end if
    ranking $\boldsymbol{S}_t$ by $\boldsymbol{score}$
    $\boldsymbol{S}'_t \leftarrow$ top ϵ sentences of $\boldsymbol{S}_t$
  $\boldsymbol{S} \leftarrow \boldsymbol{S}'_t$
end for
return $\boldsymbol{S}$

2.2 単語の特徴量を用いた要約手法

この手法では文の重要度を計算する際に，単語の特徴量を考える．単語の特徴量は tf-idf によって計算し，tf-idf のスコアの上位 n 単語を文書を特徴付ける重要単語とし，重要単語が多く含まれる文ほど重要とした．本手法での手順を Algorithm1(手法 B) に示す．入力は手法 A と同じものに加え，上位 n 単語を決定する n を与える．

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 1 に用いたデータセットの詳細を示す．手法 A を用いた実験 1~5 と，手法 B を用いた実験 6~8 を行う．各実験の出力文数の詳しい設定は表 2 に示す．

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

表 2:出力文数

実験 1 ~ 3，6~8		実験 4		実験 5	
元データの総文数	出力文数	元データの総単語数	出力文数	元データの総単語数	出力文数
1 ~ 100	2 文	1~1000	2 文	1~1000	1 文
101 ~ 500	4 文	1001 ~2000	4 文	1001 ~ 2000	2 文
501 ~ 1000	総文÷ 100	2001 ~ 5000	総単語÷ 500	2001 ~ 5000	総単語÷ 1000
それ以上	10 文	それ以上	10 文	それ以上	10 文

実験 1： 単語の一致による表層的意味のみを利用し，LexRank によりランキングをし (手法 A)，出力文数は総文数に比例する．

実験 2： 表層的意味と LDA[3] による潜在的意味を半分ずつ利用し，LexRank によりランキングをし (手法 A)，出力文数は総文数に比例する．

実験 3： 表層的意味 0.2，潜在的意味 0.8 の割合で利用，LexRank によりランキングをし (手法 A)，出力文数は総文数に比例する．

実験 4： 表層的意味と LDA による潜在的意味を半分ずつ利用し，出 LexRank によりランキングをし (手法 A)，力文数は総単語数に比例する．

実験 5： 表層的意味と LDA による潜在的意味を半分ずつ合わせて利用し，LexRank によりランキングをし (手法 A)，出力文数は実験 4 の半分とする．

実験 6： tf-idf による上位 1 単語の特徴量を利用し (手法 B)，出力文数は総文数に比例する．

実験 7： tf-idf による上位 3 単語の特徴量を利用し (手法 B)，出力文数は総文数に比例する．

実験 8： tf-idf による上位 5 単語の特徴量を利用し (手法 B)，出力文数は総文数に比例する．

実験 9： 手法 A において，表層的意味と潜在的意味の割合を 0.1~1.0 まで 0.1 単位で変化させる，

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 3:実験結果

	精度	再現率	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	**0.83**	0.22	0.31
実験 5	0.73	**0.31**	**0.38**
実験 6	0.75	0.24	0.33
実験 7	0.77	0.20	0.30
実験 8	0.78	0.20	0.28

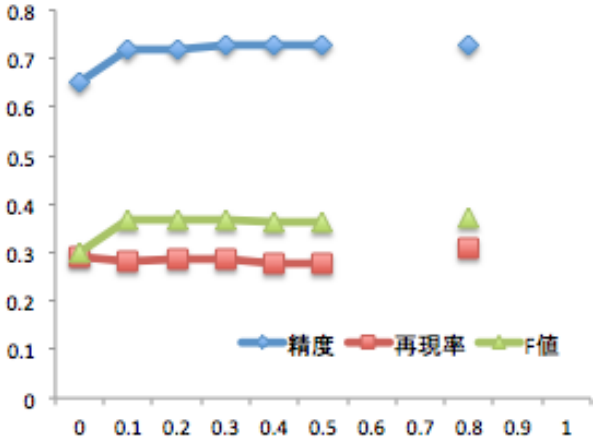

図 2: 表層的意味と潜在的意味の割合

3.2 実験結果と考察

実験 1~8 の結果は表 3，実験 9 の結果は図 2 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 1~9 はすべて性能が上回った．精度が高く再現率が低くなったのは，異なる新聞社の複数の要約文を正解データとして利用したので，正解要約とするものが本来あるべき要約文よりもサイズが大きくなっているためだと考えられる．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 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．また実験 9 の図 2 より，潜在的意味を利用することで性能は向上するが，潜在的意味と表層的意味の割合を変えることでは性能の違いは得られなかった．

また，各実験による出力文数の決定手法を比較すると，実験 2，4，5 では 2 と 5 はほぼ同じ結果となった．4 と 5 では 4 の方が精度が高く，5 の方が再現率が高くなったのは，4 の方が出力文数が多く正解文を多く含んだがその分不正解分も多く含んだからだと考えられる．これらから，この実験の結果のみからでは出力文数を元データの総文数と総単語数どちらから決定するのが良いかは判断できないので，さらに多様な設定を考えて実験する必要があると思われる．

実験 2，3(手法 A) と実験 6~8(手法 B) を比較すると，手法 A の方が性能は良かったことより，文の類似度を求める際に，tf-idf による単語の特徴量のみにより文のスコアを決定している手法 B よりも，手法 A の表層情報に加え潜在情報を用いることは有効であると言える．

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[1] よりも性能が良いことが分かった．しかし，出力文数の設定方法はさらに細かく値などを変え実験を行いよりより出力文数を探すことで更なる性能の向上が期待できる．また，単語の特徴量による重要文抽出手法は潜在的意味と表層的意味を用いた場合よりも性能が低い事より，潜在的情報を取り入れることは有用だと分かった．

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.