文の類似度グラフを用いた複数時系列文書要約

理学専攻・情報科学コース
1640641
柏井 香里

1 はじめに

ニュースや新聞記事といった時系列文書は時々刻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こと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にまとめながら，数日前には無かった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 前日との差分を用いた要約

2.1 表層および潜在情報を用いた要約

本研究では，時系列文書要約とグラフを用いた文書要約である LexRank のそれぞれの手法をふまえた時系列複数文書要約手法を提案する．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る．この手法では，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．

提案手法の概要を図 1 に示す．図 1 には 1 日前まで遡った時の，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．

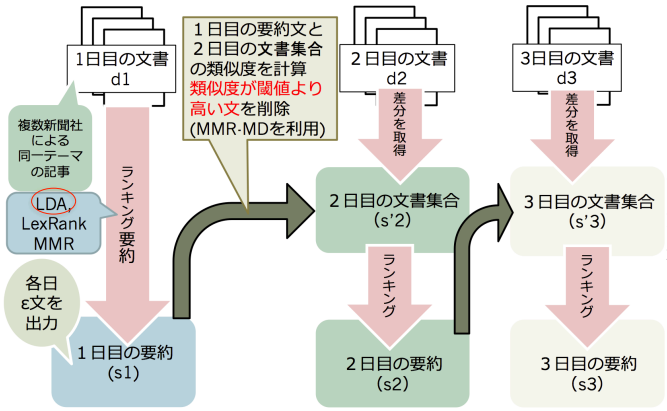

図 1: 提案手法の概要

本研究では，各文の重要度を決定するためにグラフ構造を用いる．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,，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．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 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り，文書中の単語は潜在的なトピックを持ち，同一トピックの単語は同一文書に出現しやすいと考え，そのトピックを教師なしで推定することができる．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．類似度の計算方法は以下の式 (1) に示す．

$$sim_{score} = \gamma sim_{latent} + (1-\gamma) sim_{surface} \qquad (1)$$

sim_{score} は本手法での類似度を意味する．sim_{latent} と $sim_{surface}$ は潜在情報と表層情報を意味する．

2.2 実験

対象データには，Tran ら [4] 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．評価には ROUGE[5] 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．LexRank のみを用いた場合をボーダーラインとし，出力文数や LDA を用いる割合を様々な形で設定し実験を行った．表 1 に出力文数 ϵ の設定，表 2 に LDA 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 ~ 3		実験 4		実験 5	
総文数	出力文数	総単語数	出力文数	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	総文数 /100	2001 ~ 5000	総単語数 /500	2001 ~ 5000	総単語数 /1000
1000 以上	10 文	5000 以上	10 文	5000 以上	5 文

表 2: LDA の割合の設定

実験 1	実験 2	実験 3	実験 4	実験 5
0	0.5	0.8	0.5	0.5

表 3: 実験 1~5 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	**0.83**	0.22	0.31
実験 5	0.68	**0.33**	**0.40**

2.3 結果と考察

実験 1~5 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 1~6 はすべて性能が上回った．再現率が高く精度が低くなったのは，正解要約と比較してシステムにより出力した要約文の文数が多かったからだと考えられる．各実験による出力文数の決定手法を比較すると，実験 5 がもっとも精度が高かった事から出力文数の決定方法及び文数は実験 5 が最優と考えられるが，出力文数が少ないことにより再現率は低下している．実験 4 が最も再現率が高くなったが，精度は低くなったので，出力する文数が多すぎたのだと考えられる．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．

2.4 word2vec を用いた要約

先述した手法の応用として，文の類似度を計算する際に word2vec[6] 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 2 のように計算した．

$$v(d) = \sum_{x \in d} v(x) \tag{2}$$

2.5 実験

実験は 2.2 と同様のデータを用いて行った．また，出力文数の設定は実験 1~3 と同じものとした．word2vec による単語の分散表現は 400 次元に設定し，英語 wikipedia コーパスを用いて予め学習したものを用いた．

2.6 結果と考察

実験 6 の結果は表 4 のようになった．これまでの実験 1,2 と比較しても，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，

出力文数は同じであるが，実験 7 の方が分量が半分以下だったことが分かった．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分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推測する．

表 4: 実験 6 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 6	0.35	0.38	0.28

3 おわりに

実験結果から，提案手法は既存の手法よりも F 値がすべて上回ったので性能が良い事が確認できた．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見つけ設定していく事が大切だと分かった．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] C. Lin, ROUGE: a Package for Automatic Evaluation of Summaries, In Proceedings of the Workshop on Text Summarization Branches Out, pp. 74-81, 2004.

[6] 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.