Vol.2020-DBS-171 No.2
Vol.2020-IFAT-140 No.2
2020/9/5

単語埋め込みと名詞句の共起グラフを用いた教師なしキーフレーズ抽出手法の提案

木村 優介[1,a)]　楠 和馬[1,b)]　寺本 優香[2,c)]　波多野 賢治[3,d)]

概要：近年，BERT をはじめとする事前学習済みの言語モデルの登場によって，大量の外部情報を容易に扱うことができるようになった．この言語モデルを用いた手法は，自然言語処理の他タスクと比べて精度が低いと言われているキーフレーズ抽出の分野でも有効であることが示されている． 先行研究では，キーフレーズが入力テキストを要約する語句であることに着目し，入力テキスト全体の分散表現に類似する表現を持つ名詞句をキーフレーズとする手法が提案されてきた．しかしこの手法では，キーフレーズ抽出において重要な特徴として考えられてきた各名詞句間の共起を考慮することはできない．また，各名詞句をノード，それらが共起する関係をエッジとしたグラフから，キーフレーズらしさの値を算出する研究では，名詞句自体の意味は考慮されていない．そこで本研究では，単語埋め込みと共起グラフの両方を考慮したキーフレーズ抽出手法を提案する．具体的には，入力テキストに出現する名詞句間の共起と，各名詞句と入力テキストとの意味的類似度を基にエッジの重みを算出した重み付けグラフを構築し，グラフ内の名詞句に対し，TextRank を用いてキーフレーズらしさの値を算出する．

キーワード：キーフレーズ抽出，名詞句，共起グラフ，情報検索

An Unsupervised Key-phrase Extraction Method using Word Embedding and Co-occurrence Graph of Noun Phrase

1. はじめに

キーフレーズ抽出とは，その入力テキストが議論しているトピックを要約した単語やフレーズを抽出するタスクであり [1]，その技術は，入力テキストの索引語や要約生成といった幅広い応用先が存在する [2, 3]．キーフレーズ抽出においては，キーフレーズになりやすい品詞を持つ語およびフレーズをテキストから候補語として抽出し，それらが持つ特徴量を元にキーフレーズらしさの値を算出することで，最終的にどれをキーフレーズとするかを定めている．

これまで行われてきたキーフレーズ抽出には，教師ありと教師なし双方のキーフレーズらしさの値を算出する手法が存在するが，特に近年では，大量のラベルを必要とせず入力テキストのドメインに依存しないといった観点から，教師なしの手法に着目が集まっている [4]．

その手法として一般的に知られている手法として，名詞句をノードとする共起グラフを用いた手法がある．これは，名詞句が高確率でキーフレーズとなることに着目した手法で，名詞句をノード，名詞句同士の共起関係をエッジとみなして，各候補語におけるキーフレーズらしさの値を PageRank アルゴリズムを援用して算出している．

しかしこのアプローチ単体では，入力テキストの量が少ない場合にうまく働かないことが報告されている [5]．これは，テキストから抽出できる情報が少ないため，各名詞句の関係を十分に表した共起グラフを構築できないことが原因である．

一方，もう一つの教師なしキーフレーズ抽出手法のアプ

[1] 同志社大学大学院文化情報学研究科
Graduate School of Culture and Information Science, Doshisha University, Japan
[2] 同志社大学文化遺産情報科学調査研究センター
Research Center for Knowledge Science in Cultural Heritage, Doshisha University, Japan
[3] 同志社大学文化情報学部
Faculty of Culture and Information Science, Doshisha University, Japan
[a)] kimura@mil.doshisha.ac.jp
[b)] kusu@mil.doshisha.ac.jp
[c)] teramoto@mil.doshisha.ac.jp
[d)] khatano@mail.doshisha.ac.jp

ローチとして，新たに単語の分散表現を用いた手法が出現している．こちらは前者の手法と異なり，短い入力テキストに対してもうまく対応することができているという報告がある [6].

しかし後者の手法では前者とは対照的に，入力テキストが長い場合に性能が低下する．これは，入力テキストで表現される意味が多岐に渡るため，分散表現でそれらの多義性を適切に表現することができないためであると考えられる．なぜなら，候補語からキーフレーズを選択する段階における重要な処理の一つに，候補語間における意味的類似度の算出があるが，分散表現が多義性を含んでいるために，人の感覚と乖離した類似度が算出されてしまうからである．

これら二つの手法は，いずれも候補語の出現頻度を用いた手法よりも高い精度でキーフレーズを抽出できることが報告されているため，それぞれの欠点を補うことで，より良い手法を開発できる可能性がある．そこで本研究では，共起グラフを用いた手法をベースに，分散表現を用いた手法を併せることで，短文長文いずれの入力テキストでも安定したキーフレーズ抽出精度を得ることを目指す．

2. 関連研究

本節では，共起グラフをベースとした手法である MultipartiteRank について説明する．また，事前学習済みモデルによる単語の分散表現とキーフレーズの抽出器に入力したテキスト (以後，入力テキストと呼称する) の分散表現のコサイン類似度の近さでキーフレーズを抽出する SIFRank についても説明する．

2.1 MultipartiteRank

グラフベースのキーフレーズ抽出手法は，グラフ全体から再帰的に得られる情報を元にノードの重要性を算出する手法 TextRank [7] であり，基本的には PageRank アルゴリズムを援用している．PageRank は，他のノードからのリンク数およびその重みが大きくなるほど，リンク先ノードの重要度が上がるという仮説を前提としている．一般に，あるテキスト内で重要度の高い語と強く共起する語も重要度が高い [8] と言われているため，Web ページを候補語，リンクを共起頻度に置き換えて PageRank アルゴリズムを援用することで，式 (1) に示すようにフレーズに対する重要度 $S(c_i)$ を計算している．

$$S(c_i) = (1-\lambda) + \lambda \sum_{c_j \in I(c_i)} \frac{w_{ij} \cdot S(c_j)}{\sum_{c_k \in O(c_j)} w_{jk}} \qquad (1)$$

$S(c_i)$ は入力テキスト内に含まれる全候補語の i 番目にあたる候補語 c_i の重要度，つまりキーフレーズらしさの値を表し，λ は $[0,1]$ の値を持つ減衰率を表し，デフォルトで 0.85 に設定されているが，TextRank が収束するまで実行されると，初期値の影響は受けないとされている．また $(1-\lambda)$ はノードの遷移確率を表している．さらに，候補語 c_j は候補語 c_i を指すノードの集合 $I(c_i)$ の一要素を，また候補語 c_k は候補語 c_i が指すノードの集合 $O(c_j)$ の一要素を指す．

TextRank では，あるウィンドウ内で共起する候補語同士で結んだグラフを構築し，キーフレーズらしさの値を式 (1) で算出している．しかし，この方法ではすでに抽出した候補語と類似する意味の候補語に対して高いスコアが伝播され，似た意味の候補語が多くスコアの上位に出現してしまう可能性がある．そこで MultipartiteRank では，候補語をあらかじめトピックごとに分類し，同じトピックに属する候補語同士でエッジを張らない共起グラフを構築するという工夫を行っている．

具体的にはまず，あるウィンドウ内で共起する候補語をノードとし，その出現順で完全有向グラフを構築し，エッジの重みを候補語の出現位置の逆数の総和を表す式 (2) で算出している [9].

$$w_{ij} = \sum_{p_i \in P(c_i)} \sum_{p_j \in P(c_j)} \frac{1}{|p_i - p_j|} \qquad (2)$$

ただし，入力テキスト内に含まれる全候補語集合の i 番目，j 番目にあたる候補語 c_i，c_j の出現位置 p_i，p_j は，候補語 c_i，c_j が入力テキスト内で出現した全出現位置 $P(c_i)$，$P(c_j)$ のうちの一要素である．

また，キーフレーズ抽出において入力テキストの最初に出てくる候補語ほど重要であるという事実を利用し，各トピックで最初に出現する候補語に接続するエッジの重み w'_{ij} を式 (3) で修正している [1].

$$w'_{ij} = w_{ij} + \alpha \cdot e^{\left(\frac{1}{p_i}\right)} \cdot \sum_{c_k \in T(c_j) \setminus \{c_j\}} w_{ki} \qquad (3)$$

ただし，α は重みの適合の強さを調整するハイパパラメタを表し，候補語 c_j が属するトピックを $T(c_j)$ で表し，c_k はその一要素である．

MultipartiteRank によって，類似の意味を持つ候補語が重複して抽出されることを防げるため，TextRank よりも高い精度で抽出が可能となった．一方，共起グラフのみを用いる手法は入力テキストが短くなるとエッジ数が少なくなるという特徴がある．このことはスコアの伝播を不十分にし，結果として精度の低下につながると本研究では考える．

2.2 SIFRank

SIFRank は，事前学習済み言語モデルである ELMo によって得られた単語の分散表現を用いたキーフレーズ抽出手法である [6]．手法名は，ELMo から得られた分散表

現に対し，smooth inverse frequency (SIF) [10] とよばれる処理を行い，候補語と入力テキストの分散表現を作成することに由来する．SIFRank ではキーフレーズは入力テキストを表す語であるという考えに基づき，候補語と入力テキスト各分散表現からコサイン類似度を計算し，その値をキーフレーズらしさの値として用いている．SIF では，高頻度語 (“the” や “and” など) の存在は文のトピックとは無関係であることと，文脈によって一部の単語は出現しないことを仮定として置いている．これらの仮定に基づいて，c_d をトピックとする文 s の生成確率は式 (4) になる．

$$\begin{aligned} Pr[s|c_d] &= \prod_{w \in s} Pr(w|c_d) \\ &= \prod_{w \in s} \left[\alpha f_w + (1-\alpha) \frac{\exp(v_w \cdot \tilde{c}_d)}{Z_{\tilde{c}_d}} \right] \end{aligned} \tag{4}$$

ただし，w はある単語を，v_w はその単語 w の分散表現を表す．f_w は大きなコーパスに出現するその単語の統計的確率を表す．ここで，文脈ベクトル c_d は次式の定常ベクトル c_0 と時変成分 c_d に分けられる．

$$\tilde{c}_d = \beta c_0 + (1-\beta) c_d$$

$$c_0 \perp c_d$$

トピックの最尤推定である文のベクトルは式 (5) で表現される．

$$v_s = \frac{1}{|s|} \sum_{w \in s} \frac{\alpha}{\alpha + f_w} v_w \tag{5}$$

ただし，ハイパパラメタ α は経験則に基づいて $[10^{-3}, 10^{-4}]$ が適切とされている．

SIFRank では，SIF で作成された入力テキストと候補語の分散表現を用いて，式 (6) でキーフレーズらしさの値を算出している．

$$\mathrm{SIFRank}(v_{c_i}, v_d) = \mathrm{Sim}(v_{c_i}, v_d) \tag{6}$$

$$\begin{aligned} \mathrm{Sim}(v_{c_i}, v_d) &= \cos(v_{c_i}, v_d) \\ &= \frac{\vec{v}_{c_i} \cdot \vec{v}_d}{||\vec{v}_{c_i}||\ ||\vec{v}_d||} \end{aligned} \tag{7}$$

SIFRank は従来のすべてのグラフベースの手法よりも科学論文誌の Abstract を用いた評価で最も高い精度であったと報告されている．一方で，入力テキストが長くなることで，分散表現のなかに複数のトピックが含まれる場合がある．その結果，多義語を仮定していない単純な類似度計算においては，その品質が低下し，最終的にキーフレーズ抽出の精度を低下させる原因になるとされている [6]．

3. 提案手法

本研究では，これまで重要視されてきた名詞句の共起グラフを事前学習済み言語モデルによって算出されたベクトル空間上で構築することで，名詞句と入力テキスト双方の分散表現のコサイン類似度を考慮しつつ，従来通り共起グラフベースでの重みづけに基づいたスコアリング算出が可能なキーフレーズ抽出手法を提案する．

3.1 候補語と入力テキストの類似度算出

ベクトル空間で構築した共起グラフに候補語と入力テキストの分散表現の類似度の考えを導入し，TextRank を用いて候補語のキーフレーズらしさの値を算出する．

本研究ではエッジの重みを作成する際にエッジそのものを座標化し，その分散表現を得たうえで，入力テキストの分散表現とのコサイン類似度を計算し，この類似度をエッジの重みを表現する特徴量に加える．これは，キーフレーズは対象となるテキスト全体を表現している，という考え方に基づいている．

コサイン類似度を算出する理由として，もう一つの意味的類似度を表す距離であるユークリッド距離はベクトルの大きさを加味するのに対して，コサイン類似度はその大きさを加味しないからである．エッジ上の点の決め方は複数考えられるが，いずれにしてもエッジの両端を担う二つの候補語と入力テキストとの分散表現のコサイン類似度を考慮する必要がある．エッジそのものの座標として，本研究では最も単純な方法であるエッジの中点を採用する．エッジを座標に落とし込むやり方にはいろいろなやり方が考えられるが，中点を採用した理由はエッジの両端にある二つのノードの分散表現を偏ることなく考慮することが可能だからである．エッジの中点の分散表現 v_m の算出には式 (8) を用いる．

$$v_m = \frac{v_{c_i} + v_{c_j}}{2} \tag{8}$$

ただし，v_{c_i}，v_{c_j} は候補語 c_i，c_j の分散表現を表す．

次に式 (9) で中点 v_m と SIF で構築した入力テキストの分散表現 v_d のコサイン類似度を算出し，その値をエッジの重みとする．

$$w_{ij} = Sim(v_m, v_d) \tag{9}$$

$$\begin{aligned} Sim(v_m, v_d) &= \cos(v_{c_i}, v_d) \\ &= \frac{\vec{v}_{c_i} \cdot \vec{v}_d}{||\vec{v}_{c_i}||\ ||\vec{v}_d||} \end{aligned} \tag{10}$$

3.2 同義語の集約

本研究では中点の算出段階において，候補語の分散表現を用いている．しかし，ELMo や BERT などの事前学習済みモデル が出力する単語の分散表現は文脈によって異なってしまうため，ある単語の同義語であっても文脈に応じて元の意味と遠い分散表現を獲得してしまう恐れがある．先行研究は精度向上に必要な処理として，同義語の統合処理を挙げており，本研究でもそれにならう．

本研究では，Embedding Alignment (EA) と呼ばれる考えを用いて，その候補語の分散表現を統一する [11]．その研究によると，ある非同音異義語の文脈依存の分散表現はどのような文脈にその語が出現しても一つのクラスタを形成することが明らかにされている．そのクラスタの中心である Embed Anchor [11] と呼ばれる座標を各候補語群の分散表現とすることで，同義語の集約を行う．この Embed Anchor は，その単語が持つ分散表現の総和の平均値を式 (11) で算出することで求められる [11]．

$$\overline{v}_{w_i} = \frac{1}{n} \sum_{s_j, p} v_{w_i^p}^{s_j} \tag{11}$$

ただし，v_{w_i} は対象のテキストに出現した i 番目の単語 w_i の Embed Anchor を表し，n は w_i の出現頻度を表す．また，s_j は w_i が出現する一文を表す．さらに，p はその文内における w_i の文頭からの位置に当たる．

3.3 重み付けグラフの構築

文脈依存の分散表現を Embed Anchor で統合することで，単語の分散表現に含まれる位置関係は失われてしまうため，従来の位置関係による重み付けを式 (12) で掛け合わせる．

$$w''_{ij} = w_{ij} \sum_{p_i \in P(c_i)} \sum_{p_j \in P(c_j)} \frac{1}{|p_i - p_j|} \tag{12}$$

ただし，$P(c_i)$ は候補語の集合の内，i 番目の候補語 c_i の入力テキストにおける全出現位置である．

語義が遠い語句および入力テキスト内においてよりはじめの段階で出現する語句に対しスコアを優先的に付与するため，MultipartiteRank と同様の完全有向グラフを構築する．ただし，MultipartiteRank において式 (3) で算出されている重み w に代わり，本研究では式 (12) で算出された重み w'' を利用する．新たに構築された重み付けグラフに対しては，先行研究と同様に TextRank を適用してキーフレーズらしさの値を算出する．本研究では，上位 N 語 ($1 \leq N$) をキーフレーズとして抽出した結果を考察する．

4. 評価実験

本節では，評価実験の方法とその結果，考察について説明する．

4.1 評価方法

本研究では先行研究とのキーフレーズの抽出精度を比較するため，MultipartiteRank が評価に用いたものと同じ下記のキーフレーズアノテーション済みデータセット (言語は英語に限定) に対し，評価対象の候補語の数 5，10，15 語ごとに適合率 P，再現率 R，F 値 F で評価を行う．

Inspec:
1998 年から 2002 年の間に収集された計算機科学からの 300 の科学論文の抄録（1 文書の平均単語数: 約 130 語）

SemEval2017:
493 の計算機科学，材料科学，物理学のジャーナル記事（1 文書の平均単語数: 約 180 語）

DUC2001:
308 のニュース記事（1 文書の平均単語数: 約 800 語）

SemEval2010:
ACM Digital Library に収録されている四つの異なる研究領域から抽出された 244 の科学論文（1 文書の平均単語数: 約 8,000 語）

各データセットの入力テキストの長さがさまざま用意されているため，入力テキストの長さがキーフレーズ抽出の精度にどのような影響を及ぼすかを確認する．なお，この評価実験は CPU に Intel Xeon E5-2698 2.2 GHz，主記憶に LRDIMM DDR4 256GB，GPU に 32GB NVIDIA Tesla V100 × 4，OS に Ubuntu 18.04 LTS を搭載しているワークステーション上で行い，単語の分散表現を得るための事前学習済み言語モデルは ELMo を用いた．

4.2 結果，考察

本研究と同じ共起グラフを用いた手法である MultipartiteRank との比較評価の結果，それぞれの適合率，再現率，F 値は表 1 のようになった．

この結果をみればわかるように，提案手法はいずれのデータセットに対しても，ほとんどの評価指標で MultipartiteRank よりもわずかに精度が高いことが明らかになった．一方，目標としていた論文抄録のような短い入力テキストに対する精度の向上も見られなかった．この原因として，提案手法のエッジの中点と入力テキストの分散表現間のコサイン類似度はいずれの場合も 0.8 前後の値が算出されたこと，つまり，どの候補語も入力テキストととの間に差異が見出せない意味的に近い候補語ばかりであったということを意味するが挙げられる．このため，MultipartiteRank の従来の重みに対しエッジの重みを掛け合わる処理がそれほど有効に働かなかった可能性がある．

このような結果となった理由は，分散表現を SIF を利用して構築する際，そのテキストに出現する全単語の分散表現を考慮しているため，テキスト内の候補語の中で出現傾向が他の候補語と異なる特異な表現がない限り，エッジの重み付けにおいて差がほとんど見られないからである．つまりこの結果から，エッジの重みがキーワードを抽出する手掛かりとして有効に働いていないケースが頻繁に起こりえるということを意味しており，今後先ず第一に改善すべき点であるといえる．

また，今回の評価実験の実行時間が，MultipartiteRank と比較して約 10 〜 15 倍要することになったことから，提

表 1　各種データセットに対する評価実験

N	Method	Inspec			SemEval2017			DUC2001			SemEval2010		
		P	R	F	P	R	F	P	R	F	P	R	F
5	MultipartiteRank	**0.286**	0.142	0.190	0.286	0.109	0.158	0.263	0.120	0.165	0.259	0.106	0.151
	Proposed Method	0.285	**0.145**	**0.192**	**0.305**	**0.122**	**0.174**	**0.276**	**0.146**	**0.191**	**0.290**	**0.110**	**0.160**
10	MultipartiteRank	0.236	0.233	0.235	0.237	0.181	0.205	0.213	0.194	0.203	0.216	0.176	0.194
	Proposed Method	**0.237**	**0.238**	**0.238**	**0.256**	**0.198**	**0.223**	**0.230**	**0.232**	**0.231**	**0.242**	**0.179**	**0.206**
15	MultipartiteRank	0.216	0.314	0.256	0.210	0.239	0.223	0.190	0.257	0.219	0.196	**0.237**	0.214
	Proposed Method	0.216	**0.317**	**0.257**	**0.234**	**0.257**	**0.245**	**0.211**	**0.298**	**0.247**	**0.219**	0.233	**0.226**

案手法は入力テキストが長くなるとその分だけエッジの数が増え，全エッジの中点の分散表現を算出するのに時間がかかってしまうという問題点があることが判明した．今後はその問題を解決するために，キーフレーズらしさのスコア付けの際に考慮する必要のないエッジを削除する方法を提案する必要がある．

5. おわりに

本研究では，候補語と入力テキスト間の意味的類似度と各候補語間の関係を考慮するために，キーフレーズの候補語をノードとした共起グラフを構築し，各エッジの重みにそのエッジの中点と入力テキストの分散表現のコサイン類似度を付与する手法を提案した．評価実験の結果，従来の共起グラフを用いた MultipartiteRank に比べ，僅かに高い精度での抽出が可能となった．

今後はエッジの重みに候補語の中点の分散表現を用いるだけではなく，二候補語と入力テキストの分散表現を結び，角の二等分線と二候補語を結んだ線の交点や意味的に近い候補語の分散表現を用いることで，各候補語と入力テキストとの意味的類似度をよりとらえた手法を提案する必要がある．また同時に，処理コストの軽減に関しても対処する予定である．

謝辞

本研究は日本学術振興会科学研究費助成事業基盤研究 (A) 19H01138 および基盤研究 (B) 19H04218 の助成を受けて遂行された．ここに記して謝意を表す．

参考文献

[1] Florian Boudin. Unsupervised Keyphrase Extraction with Multipartite Graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 2, pp. 667–672. Association for Computational Linguistics, 2018.

[2] Minmei Wang, Bo Zhao, and Yihua Huang. PTR: Phrase-Based Topical Ranking for Automatic Keyphrase Extraction in Scientific Publications. In *23rd International Conference on Neural Information Processing*, Vol. 9950 of *LNCS*, pp. 120–128. Springer International Publishing, 2016.

[3] Yongzheng Zhang and Nur Zincir-Heywood and Evangelos Milios. World Wide Web Site Summarization. *Web Intelligence and Agent Systems*, Vol. 2, No. 1, pp. 39–53, 2004.

[4] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, Vol. 509, pp. 257 – 289, 2020.

[5] Jiahong Li, Guimin Huang, Chunli Fan, Zhenglin Sun, and Hongtao Zhu. Key word extraction for short text via word2vec, doc2vec, and textrank. *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 27, pp. 1794–1805, 2019.

[6] Yi Sun, Hangping Qiu, Yu Zheng, Zhongwei Wang, and Chaoran Zhang. SIFRank: A New Baseline for Unsupervised Keyphrase Extraction Based on Pre-Trained Language Model. *IEEE Access*, Vol. 8, pp. 10896–10906, 2020.

[7] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411. Association for Computational Linguistics, 2004.

[8] Guangming Lu, Yule Xia, Jiamei Wang, and Zhenling Yang. Research on Text Classification Based on TextRank. In *Proceedings of the 2016 International Conference on Communications, Information Management and Network Security*, pp. 319–322. Atlantis Press, 2016.

[9] Adrien Bougouin, Florian Boudin, and Béatrice Daille. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language.

[10] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *5th International Conference on Learning Representations*, pp. 1–16. OpenReview.net, 2017.

[11] Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 1599–1613. Association for Computational Linguistics, 2019.