Vol.2018-DBS-167 No.23
Vol.2018-IFAT-132 No.23
2018/9/13

学術論文中のデータセット名の自動特定に向けて

市山 隼風[1]　後藤 七海[1]　新山 慎太郎[1]　池田 大輔[1,a)]

概要：Web 上でアクセスできる電子媒体の学術論文が急増し，これらの中に眠っている知識を発掘することが可能になってきた．しかし，論文は専門家向けに書かれており，背景知識がなしでは用語の理解さえ困難である．本研究の大きな目的は，学術論文から手法等を抽出し，二次利用を促進することであり，そのための第一歩として，データセット名の自動的な抽出を目指す．そのために機関リポジトリの学術論文を利用し，データセット名の使われ方を用いて，データセット名を特定する．提案する手法はデータセット名の候補となるパターンをヒューリスティックに特定し，データセットらしさを特定の単語との類似度で測る．従来研究と比較して，大量の学術論文を用いてトークンの分散表現を学習したモデルにより類似度を測る点が異なる．データセット名を抽出できたものの，手法やモデルなども抽出されてしまい，今後の改善が必要である．データセットが高精度で特定できるようになれば，データセットの検索エンジン等を構築でき，オープンデータやシチズンサイエンの動きに大きく貢献できると期待される．

1. はじめに

Web 上に様々な学術リポジトリが存在し，多くの学術論文に制約なくアクセスできるようになってきた．例えば，arXiv[*1]や PubMedCentral[*2]は，有名な分野リポジトリで，それぞれ主に数理物理学と生物医科学分野の論文を多く提供している．他の分野でも同様のリポジトリは存在し，例えば，RePEC[*3]は経済学，SSRN[*4]は社会学の分野リポジトリである．日本では，大学や研究機関が提供する機関リポジトリも普及しており，構築中のリポジトリもあわせれば，すでに 800 を越える数のリポジトリが存在する[*5]

このような学術リポジトリの学術論文は，基本的に制限なく利用できるため[*6]，論文をテキストデータとみなし，テキストマイニングや機械学習により，膨大な学術論文の二次利用が期待されている．実際，CORE[*7]では機関リポジトリの論文を収集し，検索エンジンとして提供するだけでなく，検索 API やダンプファイルを提供している [4]．最新版のダンプファイルは約 330GB あり，約 9.8 百万のフルテキスト論文が収録されている．

しかし，論文を読むには専門的な知識が必須であり，ある用語が何を表しているかを知ること自体が難しい．実際，研究者であっても，専門ではない分野の論文を読むことは簡単ではない．さらに，論文中に文章として明示的に書いていない背景知識が必要でもあり，論文を理解することをさらに難しくしている．逆に言えば，背景知識を共有している人同士であれば，ある程度短い分量で簡潔に主張を書くことができる利点があるが，コンピュータによる自動的な処理を困難にしているとも言える．学術論文ではないが，深層学習等の最新の機械学習技術と大量のデータを用いても，機械的に大学入試の問題を解くことが難しいのは，人間が一般的に持つ常識的な理解が機械には十分ではないからと言われている [12]．これを考えると，様々な分野で常識が異なる学術論文における背景知識を自動的に獲得することはチャレンジングな課題と言えるだろう．

本研究の大きな目的は，学術論文から手法やデータセット名等を抽出し，二次利用を促進することである．例えば，ある分野では常識的に用いられている手法が，別の分野が抱えている困難なタスクの解決に用いられるといったオープンイノベーションの創出を期待している．言い方を変えれば，自然言語で書かれ，また，分野固有の常識が背後に存在する学術論文を明示的な形に構造化（またはデータベース化）することである．そのための第一歩として，本稿ではデータセット名の自動的な抽出を目指す．オープンデータ等の動きもあわせると，様々な分野の研究者や，さらには非研究者がデータを探しやすくなることが期待される．

[1] 九州大学
Kyushu University, Moto-oka 744, Fukuoka 819–0395, Japan
[a)] daisuke@inf.kyushu-u.ac.jp

[*1] https://arxiv.org/
[*2] http://pubmedcentral.org
[*3] http://repec.org
[*4] http://www.ssrn.com/
[*5] https://www.nii.ac.jp/irp/archive/statistic/
[*6] 出版社サイトでは，提供される論文のクローリングや，ダウンロードした論文を用いた機械学習やテキストマイニングは禁止されていることも多い．
[*7] https://core.ac.uk/

ⓒ 2018 Information Processing Society of Japan

一般的なテキストマイニングでは，2 節で後述するように，情報抽出や固有表現抽出として，類似のタスクが研究されており，本研究が扱うタスクもその一部と見ることもできる．一方，分野に応じて書き方や背景知識が異なるため，ある言語に共通するパターンで抽出できるわけではない．そのため，データセット名の抽出や，データセットへのリンクの半自動的な発見，データセットの検索エンジンの構築など，近年データセットに関する研究が現れはじめた [1], [2], [7], [8].

これらの手法は，基本的にヒューリスティックを用いたパターンマッチや簡単な類似度を用いていることが多い．例えば，[8] では，一般的な英語辞書に含まれないもので，大文字のみからなるトークンをデータセット名の候補としているが，これでは ACE[*8]のように，辞書に含まれる単語を見つけることはできない．また，評価に用いたデータは非常に小さく，まだ十分な精度が得られているとは言い難い．そこで，本稿では機関リポジトリから集めた大量の学術論文を学習データとして用い，word2vec[5], [6] を用いて類似度を測る．広い範囲で候補を取る一方，word2vec を用いた類似度でデータセット名を着実に取得できると期待している．

2. 関連研究

データセット名を自動的に抽出するということは，簡単に表現すれば，特定の文脈で共通に現れるパターンを抽出するということになる．そのため，まず，広い範囲において共通なパターンを抽出する Wb からの情報抽出や固有表現抽出を説明し，次にデータセット名抽出について説明する．

2.1 共通表現やパターンの抽出

共通のパターンを探すことはテキストマイニングにおける中心的なタスクであり，対象のテキストの性質にあわせて様々なタスクが研究されてきた．対象が半構造化テキストの場合，例えばあるニュースサイトの Web ページから，共通部分であるテンプレート部分を自動的に特定して，コンテンツを自動的に抽出する情報抽出が研究されきた [3], [9]．対象が自然言語の場合，テキスト中の人名や組織名といった特定の語句を抽出する固有表現抽出が古くから研究されてきた．例えば，英語における MUC (Message Understanding Conference) や，日本語における IREX (Information Retrieval and Extraction Exercise) などで固有表現抽出タスクが行われていた．固有表現抽出タスクでは，形態素に分けた後，各形態素に対して固有表現に対応するタグを，分類器を用いて付与する系列ラベリングに基づく手法が一般的である [10], [11].

[*8] 筆者らが使ったことのある ACE は NASA により打ち上げられた人工衛星が取得するデータセットである．

固有表現はある言語に共通な表現と言えるが，同様にデータセット名はある研究分野に共通な表現と言える．つまり，前者がより広い範囲で共通表現で，後者はより狭い範囲での共通表現と言える．そのため，分野が異なれば同じ文字列でも手法の名前になることもあるだろう．一方で，情報抽出のテンプレートはサイトごとに異なるなど，より狭い範囲での共通パターンと言えるが，HTML のタグ等を含む，より長く人工的なパターンであることが多く，自動的な処理に適している．逆に，データセット名は通常単語長程度に短く，パターンそのものには多くの情報を含んおらず，その点がチャレンジングである．

2.2 データセット名抽出

データセット名の抽出は，ある種の語彙体系の自動的な構築とも言えるが，分野を問わずに語彙体系を構築する点が特徴的である．逆に，分野を固定すると，すでに存在する語彙体系を利用できる場合があり，この場合は学術論文等の二次的な利用が期待できる．

文献 [2] では，データセットを 1 つ固定して，ある論文がそのデータセットを使っているかどうかを，図表や謝辞の近くの単語を素性とした SVM により評価している．しかし，これはデータセット名を固定しているため，データセット名を抽出できない．

文献 [1] では，社会科学の分野におけるデータセットへリンクを特定しているが，データセットそのものの抽出は行っていない．

文献 [8] では，論文の「実験」「結果」「評価」などあらかじめ定めた節から大文字のみからなるトークンを抽出し候補とする．一般的に辞書にあれば候補から削除する．各候補に対し，NGD (Normalized Google Distance) を用いて，“dataset” と近い距離を持つトークンをデータセット名として出力する．これは，一般的な検索エンジンを用いて距離を求めるため，訓練例が不要という利点があるものの，候補を抽出するパターンが単純で，1 文字でも小文字が含まれていると対象からはずれてしまう．また，一般的な単語と同じ綴りを持つデータセット名も抽出できない．

3. 提案手法

文献 [8] の手法と同様，基本的には候補パターンを抽出し，“dataset” との類似度を測り，あるしきい値以上のものをデータセット名として抽出する．候補パターンとして，全て大文字とすると，小文字を 1 文字含むようなデータセット名が抽出できないため，今回はトークンの途中で小文字が 1 回は出現してよいと制限を緩くした．

このようなトークンを $ABcD$ とする．このトークンは何らかの略称であることが多いと仮定し，その近傍に `a.* b.* c.* d.*`という 4 文字のトークンの列がある場合にのみデータセット名の候補とした．近傍は，あらかじめ

ウィンドウサイズとして固定長を与え，今回の実験では 20 とした．つまり，トークンの前後それぞれ 20 トークン分が近隣である．

さらに，word2vec で作成したモデルを用いて “dataset”, “datasets”, “database”, “databases” との類似度を測り，いずれかとの類似度があるしきい値以上であった場合データセット名とする．これらの 4 つの単語は，実際に作成したモデルと，いくつかのデータセット名との距離を測り選定した．

4. 実験

4.1 実験データ

実験には，CORE[*9]が提供する 2016 年 10 月時点でのダンプファイルを用いる．CORE の Web ページによると，フルテキストを持つ約 4 百万の論文が収録され，サイズは zip 圧縮した状態で約 102GB ある．

CORE が提供するファイルに含まれているデータに，以下の前処理を施し利用する．具体的には，約物や等号，句読点など図 1 に示す python の正規表現に合致する記号を削除した後に，空白で区切ってトークン化[*10]する．

次に 1 文字目が大文字で，2 文字目以降がすべて小文字のトークンのみ 1 文字目を小文字化する．これにより，例えば “SaaS” のようなトークンはそのまま扱われるが，“This” や “We” のようなトークンは，それぞれ，“this” や “we” と変換される．

このように変換した後の異なるトークンの数は 307,288 個で，頻出な上位 20 個のトークンを以下に示す．

```
 1  1421239 the
 2  1033674 of
 3   805680 and
 4   609737 in
 5   545384 a
 6   535831 to
 7   300485 for
 8   239157 is
 9   200136 that
10   193121 with
11   179747 s
12   165337 by
13   155649 as
14   145085 on
15   140174 e
16   134784 be
17   133660 this
18   130671 n
19   129517 or
20   128017 was
```

“s” や “e” のように英単語とは限らないトークンが含まれるが，特に削除はせずにそのまま扱った．出現回数が 1 回のものは全て無視した．

改行文字等も除いたこの状態で約 300GB 程度のファイルサイズがあったが，このうち先頭 3GB を用いて学習させた．論文数としては 4518 個に相当する．

4.2 実験環境

実験に用いたマシンは Ubuntu18.04LTS が動く，Dell Precision タワー 7910 で，CPU に Intel Xeon(2.10GHz) プロセッサを 32 個を搭載し，メモリサイズは 125.8GiB である．

実験のためのプログラムは python3.6.5 で記述し，自然言語処理ライブラリ gensim と，これに含まれる Word2Vec モジュールを用いた．学習時のパラメータとしては，skip-gram モデルを用い，中間層のサイズは 200，ネガティブサンプリング数は 5，ウィンドウサイズは 10，出現回数が 1 回以内の単語を無視するようにし，32 並列で処理するようにした．

4.3 実験結果

既存の研究と異なり，特定の分野に限定しているわけではないため，どのように評価するかが大きな問題となる．実際，主に大文字からなる何文字かのトークンだけでデータセット名かどうか判断することは非常に難しく，正確を期するなら，もとの論文にある程度目を通す必要があるだろう．

今回は，類似度が高い順に上位 20 件を，出力されたトークンとその近隣の文脈と共に出力し，提案手法の有効性を確認する．

表 1 にはサーベイやデータベース，辞書などデータセットと考えてよいものが含まれている．一方で，データセット名以外に，PCA や VGAM のように手法やモデル，あるいは組織の名前が抽出されていることが分かる．データは手法やモデルと共に使われることを考えると，このような種類が上位にくることも自然である．word2vec は加法性を持つため，データセットが持つ性質をうまく用いて，手法やモデルの類似性を低くするような演算を探す必要がある．

5. おわりに

本稿では，学術リポジトリ上で提供されている論文を大量に用いて論文中のトークンをベクトル化し，データセットを表す特定の単語との類似度によりデータセット名の抽出を試みた．

データセットの抽出もできたものの，データセット以外

[*9] https://core.ac.uk/
[*10] 後に具体例で示すようにトークン化しても英単語であるとは限らない．

ⓒ 2018 Information Processing Society of Japan

```
re.compile('\[|\]|\{|\}|\(|\)|=|\||\-|<|>|:|;|\"|\'|\.|,')
```

図 1 前処理で削除した記号類の正規表現

表 1 “dataset”, “datasets”, “database”, “databases” のいずれかと類似度が高いトークン 20 件，それらの類似度及び正式名称と思われるトークン近隣の文脈

トークン	類似度	トークン近隣の文脈
MSLA	0.7069758	’mapped’, ’sea’, ’level’, ’anomaly’
CBIR	0.6979505	’content’, ’based’, ’image’, ’retrieval’
PCA	0.6379499	’principal’, ’components’, ’analysis’
VGAM	0.6051862	’vector’, ’generalized’, ’additive’, ’model’
KDHS	0.5983720	’kenya’, ’demographic’, ’health’, ’survey’
BLASTP	0.5928094	’basic’, ’local’, ’alignment’, ’search’, ’tool’, ’program’
GIS	0.5722876	’geographical’, ’information’, ’system’
ORES	0.5683997	’ORANI’, ’regional’, ’equation’, ’system’
SESA	0.5669162	’SGA’, ’experiment’, ’set’, ’analyser’
SCFD	0.5655370	’semantic’, ’closeness’, ’from’, ’disambiguity’
CYGD	0.5646238	’comprehensive’, ’yeast’, ’genome’, ’database’
CoPS	0.5627788	’centre’, ’of’, ’policy’, ’studies’
EBI	0.5610322	’ergosterol’, ’biosynthesis’, ’inhibitors’
IASCF	0.5582382	’international’, ’accounting’, ’standards’, ’committee’, ’foundation’
OPN	0.5544391	’office’, ’productivity’, ’network’
OALD	0.5532236	’oxford’, ’advanced’, ’Learner’ s’, ’dictionary’
KIMR	0.5480386	’kermadec’, ’islands’, ’marine’, ’reserve’
NVL	0.5462620	’northern’, ’victoria’, ’land’
NMRFS	0.5445106	’national’, ’marine’, ’recreational’, ’fishing’, ’survey’
LARALL	0.5428956	’low’, ’application’, ’rate’, ’and’, ’low’, ’labour’

でもデータとよく共起するような種類のトークン，例えば，手法の名前が抽出されていることが分かった．word2vec の利点として，“king-man+woman” のような演算ができ，これによって “queen” が得られる．提案手法では，まだこのような演算を用いておらず，様々な分野のデータセットをうまく抽出するような語句や演算を探す必要がある．あるいは，文献 [2] で用いられたように，図表のキャプションや謝辞の部分に着目することも考えられる．例えば，謝辞に手法の名前が入るとは考えにくい．

分野を問わない大量のデータを用いた副作用として，定量的な評価が困難になったため，今後の課題としてデータセット名の正解リストを作成するなど，評価方法を確立することが重要である．

謝辞 本研究は科研費 15H02787 の助成を受けたものです．本研究には CORE[*11] が提供する学術論文データを利用した．

参考文献

[1] Ghavimi, B., Mayr, P., Vahdati, S. and Lange, C.: Identifying and Improving Dataset References in Social Sciences Full Texts, *ArXiv e-prints* (2016).
[2] Ikeda, D. and Seguchi, D.: Automatically Extracting Keywords from Documents for Rich Indexes of Searchable Data Repositories (2017).
[3] Ikeda, D. and Yamada, Y.: Gathering Text Files Generated from Templates, *Proceedings of VLDB Workshop on Information Integration on the Web*, pp. 21–26 (2004).
[4] Knoth, P. and Zdrahal, Z.: CORE: Three Access Levels to Underpin Open Access, *D-Lib Magazine*, Vol. 18, No. 11/12 (online), DOI: http://doi.org/10.1045/november2012-knoth (2012).
[5] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *CoRR* (2013).
[6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems 26* (Burges, C., Bottou, L., Welling, M., Ghahramani, Z. and Weinberger, K., eds.), Curran Associates, Inc., pp. 3111–3119 (online), available from ⟨http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf⟩ (2013).
[7] Singhal, A., Kasturi, R. and Srivastava, J.: DataGopher: Context-based Search for Research Datasets, *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration*, pp. 749–756 (online), DOI: http://doi.org/10.1109/IRI.2014.7051964 (2014).
[8] Singhal, A. and Srivastava, J.: Data Extract: Mining Context from the Web for Dataset Extraction, *International Journal of Machine Learning and Computing*, Vol. 3, No. 2, pp. 219–223 (2013).
[9] Yamada, Y., Ikeda, D. and Hirokawa, S.: , *Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*, pp. 332–339 (2002).
[10] 中野桂吾，平井有三：日本語固有表現抽出における文節情報の利用，情報処理学会論文誌，Vol. 45, No. 3, pp.

[*11] https://core.ac.uk/

ⓒ 2018 Information Processing Society of Japan

934–941 (2004).
[11] 山田寛康, 工藤拓, 松本裕治: Support Vector Machine を用いた日本語固有表現抽出, 情報処理学会論文誌, Vol. 43, No. 1, pp. 44–53 (2002).
[12] 新井紀子: 「東ロボくん」がぶつかったロングテールの壁, 日本経済新聞, (オンライン), 入手先 ⟨https://www.nikkei.com/article/DGXKZO10389010X01C16A2X12000/⟩ (2016).

ⓒ 2018 Information Processing Society of Japan