International Journal of Institutional Research and Management
International Institute of Applied Informatics
2020, Vol. 4, No. 1, 17 – 30

Automatic Identification of Dataset Names in Scholarly Articles of Various Disciplines

Daisuke Ikeda [*], Kota Nagamizo [*], Yuta Taniguchi [*]

Abstract

Although the number of freely accessible scholarly articles is increasing, it is difficult for non-experts to understand them since they are written for experts and require background knowledge. Our big goal is to facilitate open innovation based on scholarly articles, developing methods to automatically extract essential elements in them. Once we could understand articles, they would be primary resources for institutional research. To this end, this paper is devoted to developing automatic identification of datasets in articles. Because a dictionary of datasets is necessary for evaluation, existing methods focused on some specific discipline. To achieve applicability to any disciplines, a machine learning approach with huge amounts of papers is adopted. Treating papers in multi-disciplines, the authors are not familiar with all dataset names in them. Therefore we quantitatively evaluate experimental results with precision@N, which does not require to know all the datasets in the papers, and qualitatively check if candidate tokens are dataset names or not using a GUI tool we have developed. Experimental results show precision@N is 0.450 and nDCG is 0.458. However, outputs include names of methods and software. It is an important future work to remove these noise tokens.

Keywords: scholarly repository, dataset name identification, vector representation, precision@N

1 Introduction

Scholarly repositories are databases of scholarly articles, and the number of scholarly repositories is drastically increasing. In fact, there exist more than two million full texts on about 800 institutional repositories in Japan, as of March, 2019 [1]. In addition to institutional repositories, there also exist many disciplinary repositories, mainly maintained by scholarly communities, such as arXiv [2], PubMed [3], RePEC [4], and SSRN [5].

Thanks to recent wide-spread of scholarly repositories, we can freely access to a huge number of scholarly articles on them. Since scholarly papers contain some knowledge about a discipline, we can expect to facilitate open innovation, reusing these papers and combining knowledge among different disciplines.

[*] Kyushu University, Fukuoka, Japan

However, it is difficult in general for general public or even researchers in other disciplines to understand scholarly articles since they are basically written for domain experts. Nevertheless, the authors believe that we can develop automatic ways to give scholarly papers a structure because scholarly papers have some format, which make domain experts read papers in their domain quickly.

The big goal of our research is to facilitate open innovation through freely available scholarly papers, developing methods to extract essential elements of them, such as materials and methods. Once we obtain such elements, combining ideas, methods, or materials in different disciplines can lead to open innovation.

From the viewpoint of institutional research, data on institutional repositories, such as the number of published papers or the number of accesses to them, can be a good resource to evaluate researches in institutions because an institutional repository is managed by an institution, such as a university, to provide scholarly contents created by members of the institution. For example, access log data of an institutional repository shows that many non-researches have accessed to contents on an institutional repository [6]. Such a fact reveals that scholarly papers, which are usually considered for researchers, are appealing to non-researchers.

Institutional research is basically conducted based on statistical data, or data collected with original systems [7],[8]. In addition to statistical data on repositories, the authors believe that contents on them, scholarly papers, would be great resources for institutional research once we could understand scholarly articles automatically because scholarly papers are direct outcomes from research. With such resources, for example, a research administration office of an institution could facilitate collaborative research projects within the same institution.

Toward the goal, the authors consider the following task: given a set of scholarly articles, automatically identify dataset names in them. The authors chose this task because it seems easier, compared to identify any concepts in text data, and it will be important for rich indexes at data repositories to identify dataset names [9].

In the field of text mining or natural language processing, similar tasks have been studied as information extraction or named entity extraction (see Section 2), and our task could be considered as a subtask of them. Methods based on these existing tasks assume that given text data is written in one language and there exist some common patterns in the text data. However, now we are considering scholarly articles of various disciplines, we can not assume some common patterns because writing styles or vocabularies differ depending on disciplines even if papers are written in one language, English.

Some researches about datasets have emerged recently, such as dataset name extraction, discovery of links to datasets, and dataset search engines [9][10][11][12]. Existing methods about datasets have focused on a small set of papers in some specific discipline. But, when we want to use identified dataset names for open innovation or collaborative research, it is crucial for such methods to be applicable to many disciplines. Therefore the authors consider dataset names in various disciplines and use papers collected by CORE [13], which collects papers on institutional repositories and provides search APIs, dump files, and search facility for collected papers [14].

Existing methods for dataset name identification use two steps of candidate-generation and similarity check. For example, in [12], tokens of capital letters are extracted as candidates for dataset names and then they are removed if contained in a standard English dictionary. Next candidate tokens are extracted as dataset names if they have high similarities with word “dataset”. The limitation on the candidate-generation step is so strict that

there are unextracted dataset names, such as ACE, which is a dataset obtained with satellites by NASA, while the similarity check is based on a loose measure since they have used a general search engine to measure similarities. The authors use a looser pattern to extract candidate tokens while more strict similarity check using vector representation of words, word2vec [15][16].

Because we treat papers in multi-disciplines, including those the authors are not familiar with, it is challenging how to evaluate experimental results. To solve it, the authors quantitatively evaluate experimental results with precision@N and nDCG, where precision@N is a popular measure for information retrieval systems [17] and nDCG (normalized Discounted Cumulative Gain) is for recommendation systems [18]. Both measures can be calculated by the top N list of outputs. To check if candidate tokens of the top N list are dataset names or not, the authors use a GUI tool they have developed. The GUI tool is developed with standard Web technologies, such as CSS, you can use it on a Web browser.

Checked with the top 20 of high similarity values, the result shows that precision@N is 0.450 and nDCG is 0.458. These figures are lower, compared to existing research. This is mainly due to our input articles are not restricted to one discipline.

This paper is an extended version of our conference paper [19], where some major differences include (1) a larger dataset of 330GB with 9.8M full text papers is used, compared to one of 102GB with 4M papers; (2) a larger subset of papers of 100GB full text papers is used to construct a model of word2vec, compared to 3GB; (3) a much more looser pattern is adopted to find candidate tokens; (4) description of quality for metadata of papers obtained from scholarly repositories is added; (5) for evaluation, nDCG is also used, in addition to precision@N; and (6) compared to the previous result 0.350 of precision@N, we obtained 0.450 due to a larger set of scholarly articles.

2 Related Work

We can consider identification of dataset names as common pattern extraction. In other words, the authors adopt an unsuprvised approach, instead of supervised ones like [20]. So, first, we explain information extraction from the Web and named entity extraction as common pattern extraction, and then related work on dataset name identification.

2.1 Common Pattern Extraction

Finding common patterns is a central task of text mining and many tasks to find common pattern have been studied, depending on properties of input texts. In case of semi-structured data, news articles are extracted from Web pages at a news site after identifying common templates of the site as common patterns [21][22]. In case of natural language texts, named entity extraction have been extensively studied. In this task, the person names, organizations, locations, etc., are called named entity.

Named entities are common expressions to one language, and dataset names are also common expression to one discipline. In this sense, the former task treats expressions distributed in a wider area while the latter ones distributed in a narrow area. This means that, in case of dataset name identification, there are many local expressions depending on disciplines, and thus it is more challenging. In case of information extraction from the Web, although each site has its own template and there are a lot of local templates, these templates are basically written in artificial languages, such as HTML, which are suitable to automatic processing.

2.2 Dataset Name Extraction

Assuming one fixed dataset, Ikeda and Seguchi developed a classifier based on support vector machines, which classifies if a given paper uses the dataset or not [9]. They used words appeared near figures, tables, and acknowledgment as features to train the classifier. It can check if one dataset is used in a paper or not. But, this method fixed one dataset name and thus is not applicable to our task.

In [10], Ghavimi et al. tried to identify links to datasets of social science, but not to extract dataset names.

In [12], Singhal and Srivastava extracted tokens of capital letters from predefined sections, such as "Experiments" and "Results". The authors think that this assumption holds because the authors use papers of only one discipline. Using pattern matching, extracted tokens were deleted if they appear in standard English dictionaries. For each candidate of extracted tokens, Singhal and Srivastava calculated the similarity with "dataset", based on NGD (Normalized Google Distance) [23]. To calculate similarities based on NGD, Singhal and Srivastava used a standard search engine. Therefore, costly preprocessing or training examples are not necessary. However, since candidate tokens must consist of capital letters and not be standard English words, there must exist many dataset names not extracted as candidates by their approach.

In addition to that, we can not expect that extracted dataset names would be useful for open innovation because Singhal and Srivastava evaluated using only 50 papers. Even in [24] by the same authors, only 400 papers were used to evaluate their proposed method.

3 Materials and Methods

3.1 Methods

Like the approach in [12], we also first generate candidate tokens and then check if, for each candidate, the similarity between words, such as "dataset" and "database".

We use two settings for candidate generation: strict patterns and loose ones, where, in the former setting, a candidate token can contain at least one lower-case letter and some symbols, including "-", "+", and in the latter case, a candidate token must contain at least one upper-case letter. It is noteworthy that this strict pattern generation is less strict than the candidate generation in [12], where only upper-case letters are allowed in a candidate token.

Let *ABcD* be such a token. We assume that a token is an abbreviation of some proper name and the abbreviation, that is the token, first appears near the proper name. Therefore, we use the following regular expression to find its proper name in case of the token: "`a.* b.* c.* d.*`". Using a window size as a parameter for being neighbor, we search a proper name in the window size around candidate token. In this experiment, we set 20 for the window size, that is we search 20 tokens for proper names before or after each candidate token.

For each candidate token, we calculate the similarity value of the token with one of the following words using a trained model of word2vec: "dataset", "datasets", "database", or "databases", where cosine similarity is used to measure similarity between two tokens. These words are chosen after some preliminary experiments. Given some threshold for similarity, similar tokens are extracted as dataset names.

Table 1: The numbers of papers with some values and no values for each metadata type

key	#valid	#null
abstract	11229482	5676916
authors	14617965	2288433
contributors	33329883	13573415
coreId	16906398	0
datePublished	14791413	2114985
doi	2574246	14332152
downloadUrl	2087088	14819310
enrichments	16906398	0
fullTextIdentifier	5262691	11643707
identifiers	656535	16249863
journals	2443697	14462701
language	644301	16262097
oai	15277463	16288935
pdfHashValue	2076865	14829533
publisher	10137080	6769318
raeRecordXml	16905316	1082
relations	7424331	9482067
subjects	15669487	346911
title	16714395	192003
topics	11595442	5310956
year	14717716	2188682

3.2 Data

We use the dump file provided by CORE [13], as of March, 2018. According to the Web page for the dataset, the size of the latest dump file in zip format is about 330GB, containing more than 9.8 million full text papers. After removing some papers which can not be read successfully, the total size is 576GB of 7.7 million full text papers.

CORE collects papers on institutional repositories and thus we can obtain papers in various disciplines, unlike disciplinary repositories. However, in general, the quality of metadata on institutional repositories is not good. To see this, the authors checked if a field of metadata has a value or not, using about 16.9 million papers sampled from 123 million metadata-only papers in the dump file.

Table 1 shows a metadata type (key), the number of papers which contain some values (#valid) and no values for the metadata type (#null). For example, 11,229,482 papers have some values in "abstract" but 5,676,916 papers do not have in this field. The authors believe that "journals" and "topics" would be useful to detect a discipline of the paper, but most of the papers do not have valid values in these fields.

As described before, many papers do not have valid values for "language" field. Therefore, as preprocessing, first the authors selected papers written in English, using langdetect module of Python. For a document d, function detect langs(d) in the module returns a list of probabilities of detected languages. For example, if we obtaine the following list:

```
[sw:0.7142823673474135, lt:0.2857133611277489],
```

then we consider that the paper is written in Swahili in about 71% and in Lithuanian in

Table 2: Top 20 frequent tokens from the papers used in this paper

rank	frequency	token	rank	frequency	token
1	301,694,228	the	11	32,762,908	as
2	192,812,517	of	12	28,856,878	by
3	148,652,763	and	13	26,770,867	on
4	116,120,727	in	14	26,428,859	this
5	101,235,862	to	15	26,165,272	was
6	78,336,746	a	16	25,276,836	are
7	52,734,608	is	17	24,695,921	be
8	49,865,713	for	18	22,296,880	from
9	40,890,890	that	19	20,089,105	were
10	36,432,055	with	20	19,473,401	at

Table 3: Top 20 frequent tokens from the smaller set of papers we used before

rank	frequency	token	rank	frequency	token
1	1,421,239	the	11	179,747	s
2	1,033,674	of	12	165,337	by
3	805,680	and	13	155,649	as
4	609,737	in	14	145,085	on
5	545,384	a	15	140,174	e
6	535,831	to	16	134,784	be
7	300,485	for	17	133,660	this
8	239,157	is	18	130,671	n
9	200,136	that	19	129,517	or
10	193,121	with	20	128,017	was

about 29%. We choose a document if the probability for English is equal to or greater than 90%, that is en ≥ 0.9.

Next the authors removed the following symbols:

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

After removing these symbols, they obtained tokens by separating with space. Then they converted digits into "0".

Next, for each token, the authors translated the first capital letter of the token into the corresponding lower-case letter if the succeeding letters of the tokens are lower case. For example, "SaaS" is treated as it is while "This" and "We" are translated into "this" and "we", respectively.

Table 2 shows the top 20 of frequent tokens after the preprocessing, where each line shows the rank of a token, its frequency, and the token. In this list, we find many stop words, such as "the" and "of".

In Table 3, for comparison, the authors show the top 20 frequent words from the smaller set of papers we used in [19]. We see similar words in both lists but frequencies are much smaller than those in the list of Table 2.

The file size of papers after preprocessing is about 298GB, which is still too large to train a word2vec model. So we use 100GB tokens for training, which corresponds to 505,280 papers.

3.3 Setting

Dell Precision Tower 7910 with 32 Intel Xeon(2.10GHz) as CPUs and 125.8GiB main memory, running Ubuntu18.04LTS, was used for experiment.

Programs were written in Python3.6.9, using Word2Vec module in gensim, a natural language processing library. As training parameters, skip-gram model was used, where the dimension size was 100, the number of negative sampling 5, the window size 10, infrequent words whose frequencies are equal to or less than 4 were removed, and we used 32 parallel processing.

4 Experiments

4.1 Evaluation

Unlike existing methods, the authors do not assume some specific discipline and there do not exist any datasets or dictionaries of dataset names, as far as the authors know. Therefore it is challenging how to evaluate results. In particular, our target is abbreviations, consisting of several letters. Therefore, it is difficult to check if given abbreviations are dataset names or not unless we are familiar with them in advance. Although we can make full use of corresponding proper names for abbreviations, we have to check the context in which some abbreviation is used.

For this purpose, the authors developed a GUI tool, which shows, given a query of regular expression, matched patterns and their neighbors in original texts (see Figure 1). In this example, the given query is "\WCVD\W", which means "CVD" surrounded with word-delimiters "\W", where "\W" is a special symbol of regular expression. It matches, for example, "(CVD)" and "-CVD "[1], but not "ECVD)" because "E" is not a word-delimiter.

In information retrieval, precision and recall are typical measures to evaluate the whole output of an information retrieval system. Precision (resp. recall) is the ratio of correctly retrieved tokens to retrieved token (resp. relevant tokens, that is dataset names) [17]. However, to calculate these measures, we need to know all dataset names in advance. The above GUI tool can be used to evaluate each tokens, but not for the whole output of a dataset name identification system. Therefore, instead of precision and recall, we use precision@N, which is used with a ranked list and defined as precision at top N tokens. In this case, we can check if a token in the top N tokens is a dataset name or not.

In addition to precision@N, we also calculate nDCG, normalized Discounted Cumulative Gain [18], which is used to evaluate ranked outputs, such as recommendation. In particular, nDCG is often used for outputs of items w_i with relevance score $\mathrm{eval}(w_i)$. Using $\mathrm{eval}(w_i)$, DCG for top k ranking is defined by

$$\mathrm{DCG} = \sum_{i}^{k} \mathrm{eval}(w_i)/\log_2(i+1),$$

while the ideal DCG, denoted by DCG_I, is defined by

$$\mathrm{DCG}_I = \sum_{i} 1/\log_2(i+1).$$

[1] Note that a space follows "CVD".

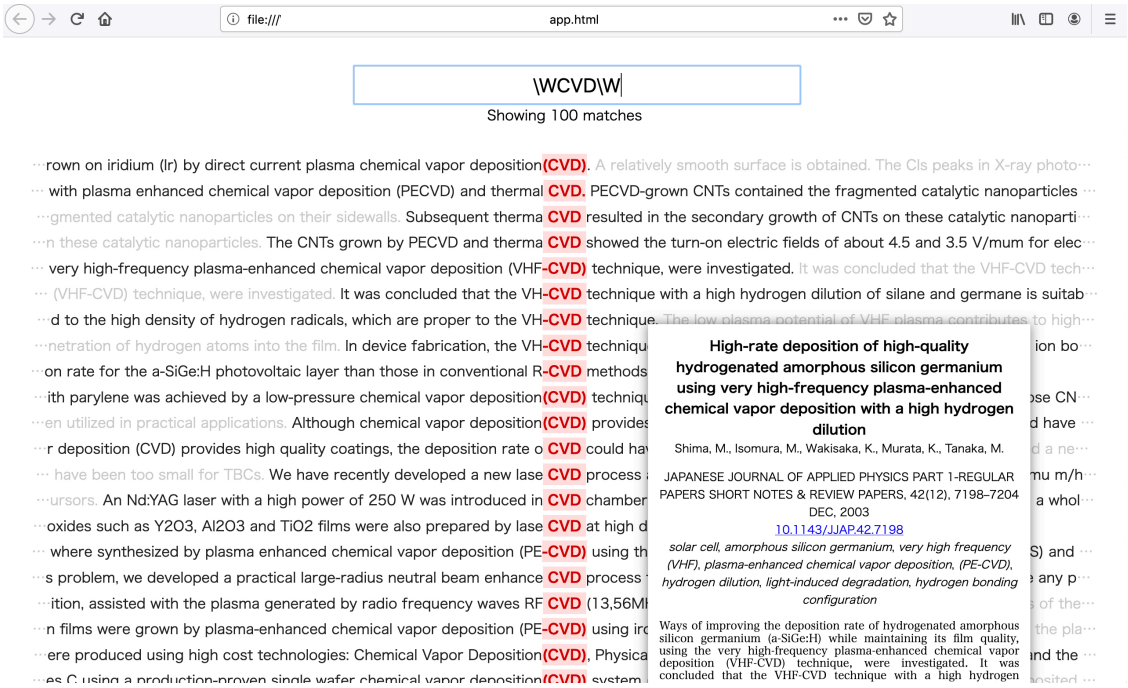

Figure 1: Screenshot of the developed GUI tool, given "\WCVD\W", which means CVD surrounded with word-delimiters (\W), as a regular expression. The tool shows matched results.

With DCG and DCG_I, nDCG is defined as follows:

$$nDCG = \frac{\mathrm{DCG}}{\mathrm{DCG}_I}.$$

A value of nDCG is between 0.0 to 1.0, and reaches to the maximail value if the ranked output is equal to the ideal ranking. Compared to precision@N, a value of nDCG is sensitive to the ranking of outputs.

In our experiments, we set $k = 20$ and, for the ith element w_i in the top k of an output, we use

$$\mathrm{eval}(w_i) = \begin{cases} 1 & w_i \text{ is a dataset name,} \\ 0 & \text{otherwise.} \end{cases}$$

4.2 Results

The authors created two models of word2vec, using 3GB and 100GB text data. To show some differences of the data size, first we show the result from 3GB data. In case of 3GB data, only strict pattern generation is used, where a pattern may contain one lower-case letter.

Table 4 shows the top 20 tokens of high similarity values, where the first column is for token, the second one for its similarity value, the third one for our judgment, where 1 denotes the authors found it as a dataset name. From original texts of the GUI tool, the authors

Table 4: Top 20 tokens which have highest similarities to "dataset", "datasets", "database", or "databases", their similarities, and our judgment, calculated using the strict pattern generation and the model tained with 3GB text data.

token	similarity	judgment
MSLA	0.7069758	1
CBIR	0.6979505	0
PCA	0.6379499	0
VGAM	0.6051862	0
KDHS	0.5983720	1
BLASTP	0.5928094	0
GIS	0.5722876	0
ORES	0.5683997	0
SESA	0.5669162	0
SCFD	0.5655370	0
CYGD	0.5646238	1
CoPS	0.5627788	0
EBI	0.5610322	1
IASCF	0.5582382	0
OPN	0.5544391	0
OALD	0.5532236	1
KIMR	0.5480386	1
NVL	0.5462620	0
NMRFS	0.5445106	1
LARALL	0.5428956	0

judged 7 out of 20 are dataset names and thus precision@N is 0.350. However, methods or models, like PCA and VGAM, are also included. We also find names for organization.

In this table, the authors only see tokens with upper-case letters although one lower-case letter is allowed in tokens. Our method allows one lower-case letter while the existing method allowed only upper-case letters in tokens. However, this result means this relaxation might not be enough.

From this table, we find that these tokens have various values of similarity, ranging from 0.7 to 0.54. This fact means that tokens at lower ranks are not so close to "database" or "dataset". Therefore, we can not conclude it has high quality with this model.

Next we move to the result from the larger size of text data. Table 5 shows the top 20 tokens which have highest similarity values, where these similarity values are calculated using the strict pattern generation and the model tained with 100GB text data. In this list, we see one token, "iHMS", has one lower-case letter in it.

Thanks to the size of text data, we also find that tokens in this list have much higher similarity values, ranging from 0.78 to 0.72, compared to Table 4.

The authors judged 9 out of 20 are dataset names, that is precision@N is 0.450 and nDCG is 0.505.

Next let us see effect of pattern generation condition. Table 6 shows the top 20 tokens which have highest similarity values, where these similarity values are calculated using the loose pattern generation and the model tained with 100GB text data. In this condition of pattern generation, if at least one upper-case letter is included in a token, we treat the token

Table 5: Top 20 tokens which have highest similarity values to “dataset”, “datasets”, “database”, or “databases”, their similarity values, and our judgment, calculated using the strict pattern generation and the model tained with 100GB text data.

token	similarity	judgment
GPKB	0.789373457	1
GEOGLE	0.770034313	0
iHMS	0.751895964	1
OBDA	0.751499474	0
PHIDIAS	0.749419689	1
REGSTATTOOLS	0.747586489	0
GPMDB	0.743063569	1
GMQL	0.742300868	1
POINeT	0.742185473	0
KUPKB	0.741823316	1
MITAB	0.738892317	0
VIRTUAL2D	0.738280594	0
DMRR	0.735069335	0
HCDN	0.734469473	1
SBAL	0.733836412	0
BICOMB	0.728748083	0
SABINET	0.727956653	1
SQIV	0.726967931	0
COPASAAR	0.725873709	1
CYTOSCAPE	0.725675702	0

as a candidate for dataset names. We can see that this relaxing leads to many tokens with many lower-case letters, such as “Pro-Cite” and “ProteinQuest”.

The authors also judged 9 out of 20 are dataset names, thus precision@N is 0.450, equal to that from Table 5. However, ranking is different between Table 5 and Table 6, and thus nDCG is 0.458, which is a little smaller than nDCG of Table 5. In general, widening candidate generation pattern may increase the number of non-dataset names. In this sense, this result is not bad since values for precision@N are the same between Table 5 and Table 6.

4.3 Discussion

In the previous section, we can find many dataset names in Table 5 and Table 6 . However, more than a half of the top 20 tokens are judged as non-dataset names. Therefore, it is important to remove such non-dataset names. Using word2vec model, we can find words sharing similar contexts. In other words, if two words appear in similar contexts, they are treated as similar. Since our approach is heavily depends on the trained word2vec model, the extracted tokens must share similar contexts. The results include many names of methods and models, such as PCA, because these names and dataset names are often used in similar contexts. Thus, just using similarities between some specific words, such as “database”, is not enough to extract only dataset names.

It is well known that word vectors obtained by word2vec have additive compositionality [16]. So we can use plus or minus operations to word vectors. For example, the

Table 6: Top 20 tokens which have highest similarity values to "dataset", "datasets", "database", or "databases", their similarity values, and our judgment, calculated using the loose pattern generation and the model tained with 100GB text data.

token	similarity	judgment
GPKB	0.789373457	1
Pro-Cite	0.783878148	0
SynapticDB	0.782725871	0
ArrayWiki	0.77789259	0
GEOGLE	0.770034313	0
GenderMedDB	0.765523553	1
CrystalEye	0.764046669	1
VarElect	0.760374784	0
CasJobs	0.760158062	0
ProteinQuest	0.759543657	0
JAX-CKB	0.756258845	1
ISI	0.755305111	1
exRNA	0.755060971	1
Web-accessible	0.752642691	0
iHMS	0.751895964	1
NLM	0.751669407	1
OBDA	0.751499474	0
MetaPathways	0.750950694	0
rnaSeqMap	0.749927461	0
PHIDIAS	0.749419689	1

following equation holds:

$$\text{“King”} - \text{“Man”} + \text{“Woman”} = \text{“Queen”}$$

It it noteworthy that we did not find "Queen" from only "King", but find it from "King" with "Man" and "Woman", where "Man" and "Woman" play an role to fix a context of "King" and "Queen". In this sense, the authors believe that this property can improve results of our method. It is an important future work to find good operations to remove methods, models, or organizations from similar tokens to "database" or "dataset".

To this end, we calculated similarity values and distances between non-dataset names and dataset ones. Figure 2 shows two graphs for cosine similarity. The left-hand side figure shows values of cosine similarity between vectors $\boldsymbol{v}_0$ of some tokens obtained by a trained word2vec model. The right-handside figure also shows values of cosine similarity, however target vectors are different from the left-hand side one, where the similarity is calculated between vectors $\boldsymbol{v}$ of the same tokens obtained from the corresponding original vectors $\boldsymbol{v}_0$ minus the center $\boldsymbol{V}_C$ of the following four vectors:"database", "databases", "dataset", and "datasets". In other words, $\boldsymbol{v} = \boldsymbol{v}_0 - \boldsymbol{V}_C$.

In these matrices, values of the similarity between dataset names are shown in the second quadrant and those between non-dataset names in the fourth quadrant. And a value of the similarity is expressed with a color, where the higher value of the similarity a cell has, the darker color the cell is filled in. For example, a cell in the second quadrant shows the value of cosine similarity between a dataset name and another dataset name.

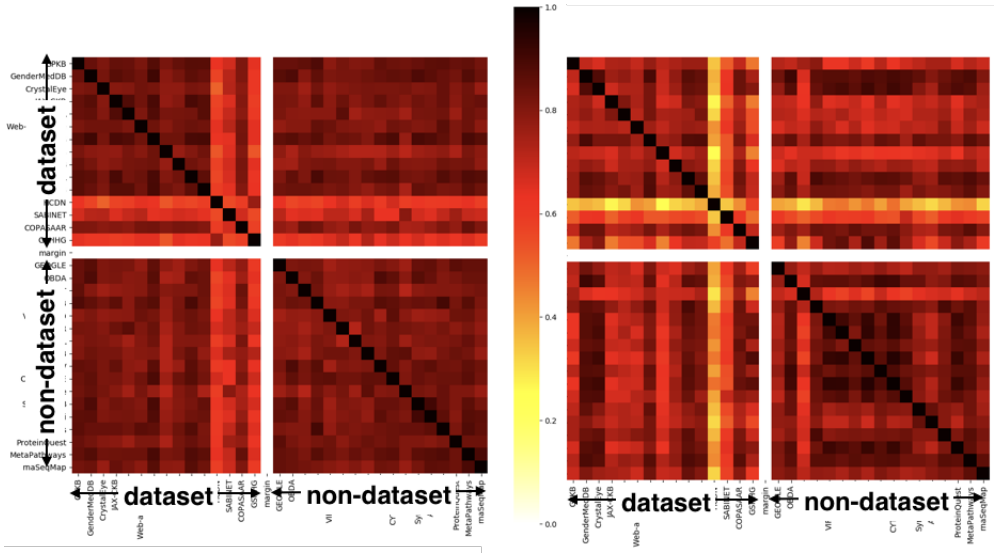

Figure 2: The left-hand (resp. right-hand) side figure shows values of cosine similarity between vectors $\boldsymbol{v}_0$ of tokens (resp. relative vectors $\boldsymbol{v} = \boldsymbol{v}_0 - \boldsymbol{V}_C$, where pairs in the second quadrant are pairs of dataset names and those in the fourth quadrant those of non-dataset names, the higher value of the similarity a cell has, the darker color the cell is filled in, and $\boldsymbol{v}_0$ denotes an original vector of the trained word2vec model and $\boldsymbol{V}_C$ the center of the following four vectors: "database", "databases", "dataset", and "datasets".

In the left-hand figure, we see that the diagonal line is black since a cell on this line is a pair of the same token and thus these cells have highest similarity values. We also see some lines in second quadrant, that is, similarity values between some specific dataset name and other dataset names have relatively low values, colored in red.

In the right-hand figure of Figure 2, we find that values in the fourth quadrant are larger (darker) than those in the same quadrant of the left-hand side figure, meaning that non-dataset names are located at similar directions from the center. We also find that values in the 2nd quadrant, pairs of dataset names, are lower than those in the same quadrant of the left-hand side figure, meaning that dataset names are located at different directions from the center. But, these facts do not mean that dataset names are far from the center non-dataset

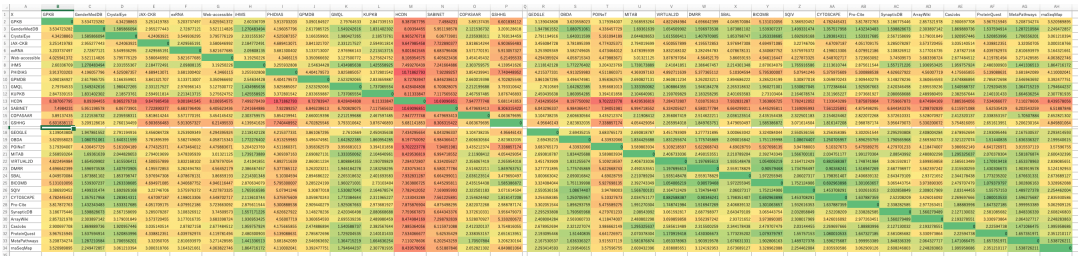

Figure 3: Distances between vectors of tokens, where the vector $\boldsymbol{v}$ of a token is obtained from the original vector $\boldsymbol{v}_0$ of the trained word2vec model minus the center $\boldsymbol{V}_C$ of the following four vectors: "database", "databases", "dataset", and "datasets", that is, $\boldsymbol{v} = \boldsymbol{v}_0 - \boldsymbol{V}_C$.

names are close to the center because the cosine similarity value for two vectors shows an angle between them.

Figure 3 shows distances, not cosine similarity, between tokens and the center. In this figure, we see that the diagonal line is green since a cell on this line is a pair of the same token and so the distance is zero. From this figure, we also see that many cells in the fourth quadrant colored in green, meaning non-dataset names are located close to each other. If we can identify these non-dataset names as one vector or in a simple notation, we can expect that they would be removed from the top output.

5 Conclusion

In this paper, the authors have proposed a method to identify dataset names in scholarly papers, and conducted experiments using real scholarly papers in many disciplines. Increasing the size of text data, the authors have improved the precision at top N list of outputs, from 0.35 to 0.45. However, there are still many non-dataset names in top N list, it is an important future work to remove these noise tokens.

References

[1] "NII Institutional Repositories Program | Documents | Statistics," https://www.nii.ac.jp/irp/en/archive/statistic/, accessed: 2020-03-04.

[2] "arXiv.org e-Print archive," https://arxiv.org/, accessed: 2020-03-04.

[3] "Home - PubMed - NCBI," https://www.ncbi.nlm.nih.gov/pubmed/, accessed: 2020-03-04.

[4] "RePEc: Research Papers in Economics," http://repec.org/, accessed: 2020-03-04.

[5] "Home :: SSRN," https://www.ssrn.com/, accessed: 2020-03-04.

[6] D. Ikeda and P. Wang, "Revealing Presence of Amateurs at an Institutional Repository by Analyzing Queries at Search Engine," in *Proceedings of the 7th International Conference of Open Repositories*, 2012.

[7] Y. Ohira, K. Ogashiwa, S. Muranaga, T. Matsumoto, and H. Naitoh, "A Questionnaire System for Institutional Research," *Information Engineering Express*, vol. 3, no. 1, pp. 9–18, 2017.

[8] K. Ogashiwa, T. Matsumoto, Y. Wang, J. Kariya, and H. Naitoh, "Evaluation of the Yamaguchi University Self-Assessment and Evaluation System and Its Improvement," *International Journal of Institutional Research and Management*, vol. 3, no. 1, pp. 1–14, 2019.

[9] D. Ikeda and D. Seguchi, "Automatically Extracting Keywords from Documents for Rich Indexes of Searchable Data Repositories," in *Proceedings of the 12th International Conference of Open Repositories*, 2017.

[10] B. Ghavimi, P. Mayr, S. Vahdati, and C. Lange, "Identifying and Improving Dataset References in Social Sciences Full Texts," *ArXiv e-prints*, 2016.

[11] A. Singhal, R. Kasturi, and J. Srivastava, "DataGopher: Context-based Search for Research Datasets," in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration*, 2014, pp. 749–756.

[12] A. Singhal and J. Srivastava, "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.

[13] "CORE – Aggregating the world's open access research papers," https://core.ac.uk/, accessed: 2020-03-04.

[14] P. Knoth and Z. Zdrahal, "CORE: Three Access Levels to Underpin Open Access," *D-Lib Magazine*, vol. 18, no. 11/12, 2012.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *CoRR*, 2013.

[17] C. D. Manning, P. Raghavan, and H. Schuetze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[18] K. Järvelin and J. Kekäläinen, "Cumulated Gain-based Evaluation of IR Techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.

[19] D. Ikeda and Y. Taniguchi, "Toward Automatic Identification of Dataset Names in Scholarly Articles," in *Developments in Open Science and Research Data Management: 8th International Conference on Data Science and Institutional Research*, 2019.

[20] K. Gábor, D. Buscaldi, A.-K. Schumann, B. QasemiZadeh, H. Zargayouna, and T. Charnois, "SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers," in *Proceedings of The 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 679–688. [Online]. Available: https://www.aclweb.org/anthology/S18-1111

[21] Y. Yamada, D. Ikeda, and S. Hirokawa, "Automatic Wrapper Generation for Multilingual Web Resources," in *Proceedings of the 5th International Conference on Discovery Science*, ser. Lecture Notes in Computer Science 2534. Springer-Verlag, 2002, pp. 332–339.

[22] D. Ikeda and Y. Yamada, "Gathering Text Files Generated from Templates," in *Proceedings of Workshop on Information Integration on the Web (IIWeb-04)*, 2004, pp. 21–26.

[23] R. L. Cilibrasi and P. M. Vitanyi, "The Google Similarity Distance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 370–383, 2007.

[24] A. Singhal and J. Srivastava, "Research Dataset Discovery from Research Publications Using Web Context," *Web Intelligence*, vol. 15, no. 2, pp. 81–99, 2017.