Classification of URLs Citing Research Artifacts in Scholarly Documents based on Distributed Representations

Masaya Tsunokake  
Graduate School of Informatics  
Nagoya University  
Furo-cho, Chikusa-ku, Nagoya, Japan  
tsunokake.masaya.z3@s.mail.nagoya-u.ac.jp

Shigeki Matsubara  
Information and Communications  
Nagoya University  
Furo-cho, Chikusa-ku, Nagoya, Japan  
matsubara@nagoya-u.jp

ABSTRACT
This paper describes methods for classifying URLs referring to research artifacts in scholarly papers, and examines their classification performance. The methods discriminate whether a URL refers to a research artifact or not and classify the identified URL into "tool" or "data." The methods use distributed representations obtained from citation contexts of the URL. Each component of a URL can be regarded as a word, and the meaning of the entire URL can be generated by synthesizing the distributed representation of each component using compositional functions. This paper evaluates several types of compositional functions from the viewpoint of classification performance. Experiments with using URLs in international conference papers showed the effectiveness of our proposed compositional functions.

CSCS CONCEPTS
• Information systems → Information extraction; Clustering and classification; Digital libraries and archives.

KEYWORDS
Open science, Data repository, Information extraction, Data citation, Scholarly document processing

1 INTRODUCTION
Open science is an activity for promoting sharing and utilizing research artifacts1. One strategy for promoting these activities is to develop and provide repositories for research artifacts. In recent years, repositories of research artifacts have been developed, such as Zenodo2 and Mendeley Data3. National infrastructures for sharing research artifacts have also been developed, such as Australian National Data Service4[28], European Open Science Cloud5[5], Research Data Shared Service6, National Data Service7[27], and NII Research Data Cloud8.

In order to establish a research artifact repository, it is required to register research artifacts and their metadata9. The number of research artifact citations in scholarly papers has been increasing in recent years. Automatically extracting information on research artifacts from a large number of scholarly papers makes the development or expansion of a repository more efficient.

This paper describes methods for classifying URLs referring to research artifacts cited in scholarly papers, all of which are extended from our previous method [29], and examines their classification performance. The methods discriminate whether a URL in scholarly papers refers to a research artifact or not, and classify identified research artifacts into the type "tool" (e.g., programs and software) or "data" (e.g., measurement data and test data).

Our previous approach uses words surrounding the URL in scholarly papers, that is, distributed representations obtained from citation contexts of the URL. The meanings of non-natural language strings such as URLs can be expressed as distributed representations. Each component of a URL, such as domain name, directory name, and file name, can be regarded as a word, and the meaning of the entire URL can be generated by synthesizing the distributed representation of each component using compositional functions.

This paper evaluated several types of compositional functions from the viewpoint of classification performance. Experiments using URLs in international conference papers showed the effectiveness of our proposed compositional functions.

2 URL REFERRING TO RESEARCH ARTIFACT
2.1 Metadata in Research Artifact Repository
Creating metadata is necessary to facilitate access to resources in repositories. The most basic metadata scheme is Dublin Core Metadata Element Set10. As an example, Table 1 shows the metadata

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>Treebank-3</td>
</tr>
<tr>
<td>contributor</td>
<td>Mitchell P. Marcus et al.</td>
</tr>
<tr>
<td>publisher</td>
<td>Linguistic Data Consortium</td>
</tr>
<tr>
<td>date</td>
<td>1999</td>
</tr>
<tr>
<td>type (DCMI)</td>
<td>Text</td>
</tr>
<tr>
<td>description</td>
<td>This release contains the following Treebank-2</td>
</tr>
<tr>
<td>identifier</td>
<td>DOI: 10.35111/gq1x-j780</td>
</tr>
<tr>
<td></td>
<td><a href="https://catalog.ldc.upenn.edu/LDC99T42">https://catalog.ldc.upenn.edu/LDC99T42</a></td>
</tr>
</tbody>
</table>

1In this paper, we denote research artifacts as materials generated or used in the course of research activities, such as tools (e.g., software, program) and data (e.g., measurement data, test data).
2https://zenodo.org/
3https://data.mendeley.com/
5https://ec.europa.eu/research/openscience/index.cfm?pg=open-science-cloud
6https://www.jisc.ac.uk/rd/projects/research-data-shared-service
7http://www.countrydata.org/
8https://rcos.nii.ac.jp/en/service/
9Information about research artifacts (e.g., name, creator, type, and usage)
10https://www.dublincore.org/specifications/dublin-core/dces/
2.2 Research Artifact Citation

Recently, research artifacts, such as datasets and software, have been increasingly cited in scholarly papers. Thus, there is a growing movement to establish formal rules for data and software citations, as FORCE11 has declared “Data Citation Principles” [6] and “Software Citation Principles” [24]. However, it is a long way off before this practice is widely spread among researchers. Howison and Bullard [8] have shown that there were many informal citations appearing in biology papers. One strategy for automatic identification of the informal citations is to identify research artifact mentions in the body text [13]. Some studies address the identification of dataset names [10, 20, 23] while others do that of software names [3, 4, 22]. On the other hand, there are many cases in which research artifacts are listed in the reference section [11] or are cited by providing the corresponding URL.

Providing URLs in papers is a common form of Web citation. Yang et al. [31] have analyzed such citations, and NLPExplorer [14] [17], which is a service for searching scholarly papers, provides access to URLs cited by the papers. We also focus on URLs in scholarly papers because many published research artifacts are accessible on the Web. However, not all URLs in scholarly papers refer to research artifacts. Therefore, we aim to identify URLs referring to the research artifacts in scholarly papers.

2.3 Classification of URLs in Scholarly Papers

Fig. 1 illustrates the design of task in our study. The goal is to identify URLs referring to research artifacts from scholarly papers and categorize them. In this task, URLs in scholarly papers are classified into the following three categories [15]:

- **tool**: program, software, toolkit, etc.
  - https://nlp.stanford.edu/projects/glove/
  - https://github.com/google-research/bert
  - http://www.nltk.org/
- **data**: observation data, experimental data, data source, etc.
  - http://qwone.com/~jason/20Newsgroups/
  - http://labelnet.org
  - http://answers.yahoo.com
- **other**: Not research artifacts (e.g., publications, services).
  - https://www.mturk.com

Our previous method [29] uses words surrounding a URL for a classifier. URLs are placed on either footnote, reference section, or body text. Even if a URL is in a footnote or the reference section, the sentences referring to the corresponding footnote or reference generally exist in the body text. For example, a footnote “http://lemurproject.org/clueweb09/*” is referred to by the following sentence in the body text [32]:

The ClueWeb09 dataset is a collection of 1 billion webpages (5TB compressed in raw HTML) in 10 languages by Carnegie Mellon University in 2009. By observing this sentence, it turns out that the above URL is provided to refer to a corpus. This paper calls one sentence referring to a URL in the body text as “citation context.”

3 URL CLASSIFICATION BASED ON DISTRIBUTED REPRESENTATIONS

A comprehensive view of all citation contexts for each URL allows us to classify it properly. Based on this idea, our previous approach [29] obtains a distributed representation of a URL from its citation contexts and uses it for classification. According to the distributional hypothesis [7], even for non-natural language strings such as URLs, their meaning could be obtained from words co-occurring in their surroundings. The following two approaches with different semantic units can be considered:

- regarding an entire string of a URL as a word
- regarding each component of a URL as a word, and obtaining the meaning of the URL from that of each component

---

11http://www.language-archives.org/
12In the field of natural language processing, the accumulation and use of the metadata of language resources have long been encouraged because the need for their repositories have been recognized [9].
13http://shachi.org/
14http://nlpexplorer.org
15Enumerated URLs in each category are examples of URLs belonging to the category.
16This URL refers to a Web page for a Q&A service. However, the scholarly papers tend to refer to the URL for pointing to a data source of question answering datasets. Therefore, in scholarly papers, the type can be considered to the “data.”
Fig. 2 shows an example for two different semantic units. The acquisition unit of a distributed representation also varies depending on the employed approach.

Nanba [16] obtained distributed representations of URLs based on the former approach. He proposed a method named “W2V-URL” of giving keywords to URLs by word2vec [15]. In W2V-URL, words whose distributed representation is highly similar to that of a URL are assigned to the URL as the keywords. We employed the URL classification based on distributed representations of URLs as baseline method [29]. The procedure of the baseline is as follows:

1. Assign a unique ID to each URL in scholarly papers and convert each URL to a tag with the corresponding ID.
2. Obtain a distributed representation for each tag.
3. Classify URLs using distributed representations.

On the other hand, we proposed the URL classification method based on the latter approach [29]. Thus, the method regards each component of a URL as a word and obtains its distributed representation. In some cases, the type of the target referred to by a URL can be inferred from the domain or directory name constituting the URL. For example, it can be inferred from the expressions of directory names “tools” and “TreeTagger” that a URL “http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger”7 points to a tagging tool. Distributed representations of these components obtained from citation contexts may be able to capture the meaning of substrings. This paper calls a component of URL as a URL element (e.g., host name, domain name, directory name, file name, and extension).

In our previously proposed method, URLs in scholarly papers are classified according to the following procedure18:

1. Decompose each URL in scholarly papers into URL elements.
2. Assign a unique ID to each URL element and convert each URL element into a tag with the corresponding ID.
3. Obtain a distributed representation for each tag.
4. Classify each URL using the vector computed by adding distributed representation of each URL element in the URL.

### 4 COMPOSITION OF DISTRIBUTED REPRESENTATIONS OF URL ELEMENTS

Our previous method classifies URLs with vectors which are computed by adding distributed representation of each URL element [29]. In this paper, some types of compositional functions for distributed representations of URL elements are evaluated.

Our previous method tends to misclassify URLs whose gold label are “tool” into the “data” class and vice versa. In addition, misclassified URLs tend to be short (i.e., URLs with a small number of directories). For example, the URL “https://twitter.com/,” which has the “data” label, was misclassified into the “tool” class. It is considered that this disadvantage is caused by URL elements with extremely high frequency, such as host and domain names. For example, URL element “com” is a generic top-level domain appearing in many URLs. The bias of citation contexts in scholarly papers might lead to characterizing the distributed representation of “com” as “tool” even though it is not critical evidence in the URL classification. The classification of short URLs especially is affected by URL elements whose frequency is extremely high.

This paper revises the step (4) in the above procedure of the previous classification method as follows:

4’ Classify a URL using the vector combined by $f(v_1, \ldots, v_n)$, where $f(\cdot)$ is a compositional function, and $v_i$ is the vector of the $i$th URL element in the URL.

We evaluate fundamental manipulations as compositional functions, such as averaging, summation, and max-pooling19. In addition, we also evaluate several functions to improve our previous method.

URL elements with extremely high frequency, such as host and domain names, are considered to be less useful for classification. To weaken the influence of frequent URL elements, we extend the fundamental manipulations by weighting with the entropy of URL elements. The entropy is computed on the basis of frequency of each URL element in scholarly papers. The entropy of each URL element was computed by

$$- \log_2 \frac{\text{Count}(w)}{\sum_{w'} \text{Count}(w')}$$

where $w$ is the target URL element, $w'$ is an arbitrary URL element in the set of all URL elements, and $\text{Count}(\cdot)$ is a function counting its argument in the scholarly papers. In addition, the top-level domains can be considered not to contribute much to the classification of the targets referred to by URLs. Therefore, we also employed manipulations except for distributed representations of top-level domain names simply.

This task can be regarded as a sequence classification task. Using a model based on recurrent neural networks (RNN) as the compositional function may realize to get better weights for synthesizing URL elements and incorporate order information into input features. Therefore, we also verify a classification method employing the gated recurrent unit (GRU) [2] as a gated RNN. Fig. 3 shows the architecture of the verified model.

### 5 EXPERIMENT

18For example, every “http://nlp.stanford.edu/software/tagger.shtml” is converted to the tag “URL2495.”

19Taking the maximum value along each dimension.
Table 2: Hyperparameters of the adopted distributed representation and classification model

<table>
<thead>
<tr>
<th>Compositional function</th>
<th>Parameters of word2vec</th>
<th>Classification model</th>
<th>Standardization</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (baseline method)</td>
<td>20 10 300</td>
<td>logistic regression with one-vs-rest</td>
<td>False</td>
</tr>
<tr>
<td>averaging</td>
<td>10 5 800</td>
<td>logistic regression with one-vs-one</td>
<td>True</td>
</tr>
<tr>
<td>summation</td>
<td>20 10 700</td>
<td>logistic regression with one-vs-rest</td>
<td>True</td>
</tr>
<tr>
<td>max-pooling</td>
<td>20 5 400</td>
<td>logistic regression with one-vs-one</td>
<td>True</td>
</tr>
</tbody>
</table>

Table 3: List of frequent URL elements in scholarly papers

<table>
<thead>
<tr>
<th>Rank</th>
<th>URL element</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>org</td>
<td>4983</td>
</tr>
<tr>
<td>2</td>
<td>com</td>
<td>3545</td>
</tr>
<tr>
<td>3</td>
<td>www</td>
<td>3505</td>
</tr>
<tr>
<td>4</td>
<td>github</td>
<td>1964</td>
</tr>
<tr>
<td>5</td>
<td>aclweb</td>
<td>1712</td>
</tr>
<tr>
<td>6</td>
<td>anthology</td>
<td>1704</td>
</tr>
<tr>
<td>7</td>
<td>edu</td>
<td>1491</td>
</tr>
<tr>
<td>8</td>
<td>doi</td>
<td>920</td>
</tr>
<tr>
<td>9</td>
<td>html</td>
<td>757</td>
</tr>
<tr>
<td>10</td>
<td>cs</td>
<td>625</td>
</tr>
</tbody>
</table>

5.1 Experimental Data

Experimental data were the same as our previous study [29]. The data were generated from scholarly papers in the proceedings of ACL 2010–2019, which are the international conferences in the field of natural language processing. Concretely, we collected PDF files from ACL Anthology [25] and converted them into texts in preserving their structural information by PDFNLT-1.0 [1]. The number of papers was 3,837. There were 12,568 URL occurrences and the number of distinct URLs was 9,480. The average number of URL elements in the URLs was 4.72, and the number of distinct URL elements was 11,724. Table 3 shows the frequent URL elements.

Many URLs are provided in footnotes or references. To capture citation contexts, these URLs were mechanically inserted into the body texts according to where the corresponding footnote or reference is referred to. After that, URLs or URL elements were converted to tags according to the procedure described in Section 3. For example, in the baseline method of our previous study [29], the citation context illustrated in Section 2.3 is transformed as follows:

The ClueWeb09 [URL2164] dataset is a collection of 1 billion webpages (5TB compressed in raw HTML) in 10 languages by Carnegie Mellon University in 2009. These processed texts of papers were used for obtaining the distributed representations.

To evaluate performances for URL classification, we labeled URLs appearing frequently in the scholarly papers with “tool,” “data,” or “other.” The created dataset contains 500 annotated URLs. The URLs described in Section 2.3 are examples extracted from this annotated dataset. The labeling ratios of “tool,” “data,” and “other” in 500 URLs are 39.8%, 33.6%, and 26.6%, respectively. Of them, 100 URLs are used as a development set.

5.2 Experiment for Basic Functions

We used word2vec [15] to obtain distributed representations and Gensim [21] for its implementation. Sentence segmentation and word tokenization were also performed by using gensim.

As the baseline, we also evaluate the classification method regarding a URL as a single word (described in Section 3). Since the baseline method does not decompose URLs into URL elements, the compositional function does not exist.

For each method, the best parameters of word2vec [21] were selected on the basis of the performance in the development set. Similarly, we also chose a classification model from logistic regression, linear SVM, and nonlinear SVM with RBF kernel, a multi-class classification approach from one-vs-one and one-vs-rest, and whether to standardize input features. Table 2 presents the selected parameters. We used scikit-learn [19] for the implementation of classifiers.

The 10-fold cross-validation was performed on the 400 URLs, excluding the development set. The development set was added to the training data for each cross-validation split. For evaluation, we computed the accuracy on the 400 URLs. We also measured precision, recall, and F1-score for each split by macro-averaging. Table 4 shows the results. The results of the averaging and summation are the best and competitive with each other.

F1-score for each label in the baseline, averaging, and summation is shown in Table 5. Compared to the baseline, both compositional

---

24Components of a scholarly paper such as title, authors, body text, figures, tables, captions, footnotes, and reference list.
25https://github.com/KMCS-NII/PDFNLT-1.0
26Strings beginning with either “http://”, “https://”, or “ftp://” were identified as URLs.
27The rates of URLs in footnotes and references are 0.767 and 0.127, respectively.
28https://radimrehurek.com/gensim/
29The epoch was 10 or 20, and the window size was 5 or 10. Dimension sizes range from 100 to 1000 in increments of 100. The other hyperparameters were the default values, except that the pruning threshold for low-frequency words was set to 3.
30https://scikit-learn.org/stable/
31Precision, Recall, and F1-score are the averages over 10 splits.
Table 5: Results of extended compositional functions and GRU

<table>
<thead>
<tr>
<th>Type</th>
<th>Composition function</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>macro average</td>
<td>tool</td>
</tr>
<tr>
<td>baseline</td>
<td>None</td>
<td>0.785 (314/400)</td>
<td>0.779</td>
</tr>
<tr>
<td>averaging-based</td>
<td>averaging</td>
<td>0.796 (319/400)</td>
<td>0.808</td>
</tr>
<tr>
<td></td>
<td>weighted by entropy</td>
<td>0.790 (316/400)</td>
<td>0.796</td>
</tr>
<tr>
<td></td>
<td>except top-level domain</td>
<td>0.788 (315/400)</td>
<td>0.799</td>
</tr>
<tr>
<td>summation-based</td>
<td>summation</td>
<td>0.800 (320/400)</td>
<td>0.808</td>
</tr>
<tr>
<td></td>
<td>weighted by entropy</td>
<td>0.798 (319/400)</td>
<td>0.805</td>
</tr>
<tr>
<td></td>
<td>except top-level domain</td>
<td>0.813 (325/400)</td>
<td>0.816</td>
</tr>
<tr>
<td>RNN-based</td>
<td>GRU</td>
<td>0.823 (329/400)</td>
<td>0.820</td>
</tr>
</tbody>
</table>

Figure 4: A heatmap of error rates for each depth of URLs

functions have the disadvantage of identifying the “tool” and “data” class. This result is the same as our previous study and it may be improved by less considering URL elements with a negative effect.

5.3 Experiment for Extended Functions

According to the results in Section 5.2, we used the averaging and summation as a basic compositional function and modified it. Therefore, the averaging and summation were extended to a weighted averaging and weighted summation by the entropy of URL elements, respectively. The entropy was computed based on frequency of each URL element in the experimental data. In addition, we also employed averaging and summation except for distributed representations of top-level domain names simply. For each function, the parameters are selected in the same way described in Section 5.2. Other experimental settings are also the same.

Table 5 shows the results. A part of extended compositional functions improved the discrimination performance of “tool” and “data” classes, which was a disadvantage of the proposed method. However, accuracy and macro-averaging F1-score of extended compositional functions are lower than that of basic compositional functions, excluding summation except top-level domain.

Fig. 4 shows error rates of each compositional function based on the summation for each depth of URLs. In Fig. 4, the higher the error rate was, the more intense the color was. As described in Section 4, summation tends to misclassify short URLs. Although the weighted summation had worse results, the performance of the summation except top-level domain was improved. This result indicates that there are frequent URL elements with useful information for the URL classification and simply excluding the top-level domains is effective for the summation. As a case study, the URL “http://www.imsdih.com/” misclassified into the “tool” class by the summation is correctly classified into the “data” class by excluding the distributed representation of “com” from the summation.

5.4 Experiment for RNN-based Function

As with the above Sections, we evaluate the classification method using GRU [2] as a compositional function. The model was implemented in PyTorch28 [18]. In the training step, we used Adam as an optimizer and cross entropy loss. In addition, dropout was applied on inputs for GRU. The weights of the embedding layer were fixed by the pre-trained distributed representations of the URL elements described in Section 5.2. The best parameters were selected based on the classification performance on the development set 29.

After setting parameters, the method employing GRU was evaluated by 10-fold cross-validation with the same setting as in Section 5.2. The experimental results are shown in Table 5. GRU outperformed other compositional functions in the accuracy and macro-averaging F1-score as well as F1-score for “tool” and “data” class which basic compositional functions had difficulty identifying. Fig. 4 shows the error rates of GRU for each depth of URLs. GRU also improved the classification performance of short URLs compared to the basic compositional function.

6 CONCLUSION

This paper described methods for classifying URLs referring to research artifacts in scholarly papers and examined their classification performance. The methods use distributed representations obtained from citation contexts of the URL. Our approach regards each component of URLs as a word, and input features for a classifier are generated by synthesizing the distributed representation of each component using compositional functions. Experimental results showed the effectiveness of our compositional functions.

ACKNOWLEDGMENTS

This research was partially supported by the Grant-in-Aid for Scientific Research (B) (No. 21H03773) of JSPS.

28https://pytorch.org/
29We trained the model employing each combination of parameters for 300 epochs and selected the best epoch. The selected dimension size of hidden state in GRU was 50 from 25, 50, 100, and 200; the selected batch size was 32 from 4, 8, 16, 32, and 64; the selected learning rate was 1.0e-3 from 1.0e-3, 1.0e-4, and 1.0e-5; the selected dropout rate was 0.2 from 0.0, 0.2, 0.4, 0.6, and 0.8; the selected epoch was 98. The epoch, window size, and dimension size of selected distributed representations of URL elements were 20, 5, and 600, respectively.