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

Masaya Tsunokake (Presenter), Shigeki Matsubara
Nagoya University, Japan

September 30, 2021
Outline

1. Introduction
2. Methodology
3. Experiments
4. Conclusion
Background

• Research Artifacts
  – digital objects created or used in the course of research work
    ▪ software, toolkits, programs, observation/experimental data
  – increasingly cited in scholarly papers
    and gathering attention as one of the research results

• Repositories for research artifacts
  – facilitate to share and utilize research artifacts
  – it is required to register metadata of research artifacts
  – metadata in the repositories make research artifacts more accessible and findable

Automatic generation of metadata
makes developing and expanding repositories more efficient
Our Vision

Introduction

repositories for research artifacts

scholarly papers

extract information about research artifact

meta data

meta data

meta data

meta data

meta data

register

repositories for research artifacts

search
Related Work

- **Automatic generating metadata**
  - Kozawa et al. [1] have proposed a method for extracting usage information from scholarly papers
    - using resource names in SHACHI [2] as clue
    - target resources were limited to ones in repositories
  - Our targets include ones not stored in existing repositories

- **Identification of citations for research artifacts in scholarly papers**
  - Some method identifies dataset [3-6] or software [3,7-9] names in the body text
  - On the other hand, there are other ways for citing them
    - listed in the reference section [10]

*Example 2* All statistical procedures were performed using IBM SPSS Statistics software version 22. Task accuracy and response times were analyzed using the SPSS software package (SPSS v17.0, Chicago, Illinois, USA).

*Example 3* This approach uses a maximum entropy classifier\(^3\) with \(L_1\) regularisation. In early experiments we found that the constituent-based approach per-

\(^3\)http://scikit-learn.org/
Introduction

Contribution

1. We proposed the methods realizing the following tasks automatically
   - identification of URLs citing research artifact in scholarly papers
   - generating information about the type of the research artifacts

   nologies, where appropriate. A video illustrating most of the user-facing features in action is currently available at https://www.youtube.com/watch?v=Efs117MWFkE

   ➢ not research artifact

   requirement is not as strict as that in human languages.

   In our experiments, we extract the antonym dictionary from the WordNet lexicon http://wordnet.princeton.edu/

   ➢ used lexicon (research artifact)

2. We evaluated the classification performances of the methods

Metadata

<table>
<thead>
<tr>
<th>title</th>
<th>WordNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>creator</td>
<td>George A. Miller, Princeton University, etc.</td>
</tr>
<tr>
<td>publisher</td>
<td>The Global WordNet Association MIT Press</td>
</tr>
<tr>
<td>type</td>
<td>Text</td>
</tr>
<tr>
<td>identifier</td>
<td><a href="http://wordnet.princeton.edu/">http://wordnet.princeton.edu/</a></td>
</tr>
<tr>
<td>usage</td>
<td>NLP, word sense disambiguation, query expansion, cluster its senses</td>
</tr>
</tbody>
</table>

*An example from [1,2]
Task Definition

- **URL classification**
  - Our goals
    - identify URLs citing research artifacts
    - detect the type of research artifacts.
  - Each URL in scholarly papers is classified based on the type of objects which the URL refers to

- **The definition of each class**
  1. **tool**: programs, software, toolkit etc.
     - [http://www.csie.ntu.edu.tw/~cjlin/libsvm](http://www.csie.ntu.edu.tw/~cjlin/libsvm) (software)
     - [https://www.tensorflow.org/](https://www.tensorflow.org/) (framework)
  2. **data**: observation/experimental data, data source, etc.
  3. **other**: not research artifacts (e.g., publications, services)
Approach

• **the Citation Context of a URL:** the corresponding sentence in the body text (referring to footnote or reference where the URL are provided)

- **intuitiveness:** reading citation contexts, we can know what resources a URL refers to
  - the system can classify an URL properly if it can captures all citation contexts of the URL

---

Methods

- The ClueWeb09 (footnote) dataset is a collection of 1 billion webpages (5TB compressed in raw HTML) in 10 languages by Carnegie Mellon University in 2009

---

We obtain **distributed representations of URLs** and use them for input features in URL classification
Distributed Representations of URLs

- two approaches to obtain distributed representations of URLs with different semantic units

**regarding each URL as a word** [11]

- This approach converts each URL to the tag and obtains distributed representations of the tags

  - The Stanford POS Tagger ([http://nlp.stanford.edu/software/tagger.shtml](http://nlp.stanford.edu/software/tagger.shtml)) is used to distinguish noun and adjective words from each other.

  - a tag (e.g., [URL930])

**regarding each component of URLs as a word** (our original approach)

- some components are considered to contain any meaning e.g., [http://trec.nist.gov/data/tweets/](http://trec.nist.gov/data/tweets/)

- This approach converts each component to the tag, obtains distributed representations of the tags, and synthesizes them for obtaining overall representations of URLs

  - The Stanford POS Tagger ([http://nlp.stanford.edu/software/tagger.shtml](http://nlp.stanford.edu/software/tagger.shtml)) is used to distinguish noun and adjective words from each other.

  - a tag (e.g., [COMP930])

- we define components as domain, directory, filename, and extension

  - we call each component **URL element**

*quoted from [25]
Methods for URL Classification

1. If each URL is regarded as a word
   - **Step 1**: Convert each URL to the tag
   - **Step 2**: Obtain distributed representations of tags (URLs)
   - **Step 3**: Classify URLs using the distributed representations as input features
   - **Step 4**: Classify URLs using the features created in Step 3

2. If each URL element is regarded as a word (proposed approach)
   - **Step 1**: Convert each URL element to the tag
   - **Step 2**: Obtain distributed representations of tags (URL elements)
   - **Step 3**: Create a feature of each URL by synthesizing distributed representations of the URL elements
   - **Step 4**: Classify URLs using the features created in Step 3

*The Stanford POS Tagger [http://nlp.stanford.edu/software/tagger.shtml](http://nlp.stanford.edu/software/tagger.shtml) is used to distinguish noun and adjective words from each other.*

*Original sentence in [25]*
Some Compositional Functions

1. **Summation** (in our previous study [12])
   - add vectors element-wise
   - overly affected by frequent URL element in scholarly papers

2. **Summation weighted by the entropy** of each URL element
   - weaken the influence of frequent URL elements
   - entropy is computed according to the frequency in papers
   
   $$-\log_2 \frac{\text{Count} (w)}{\sum_{w'} \text{Count} (w')}$$

3. **Summation except top-level domains**
   - top-level domains may be not useful for the classification
   - exclude top-level domains from the computation

4. **GRU** [13]
   - to get better weights for synthesizing
   - incorporate order information

---

Methodology

- **Embedding Layer + ReLU**
- **concat + ReLU**
- **Linear + Softmax**
- **tool / data / other**
- **GRU**
- **taku910 github io mecab**

[Diagram showing the flow of data through different layers including GRUs and ReLUs, along with URLs and text components like `taku910`, `github`, `io`, `mecab`, and a URL: `http://taku910.github.io/mecab/`.]
Experimental Setup

**Purpose:** to evaluate classification performances of the methods

**Dataset:** based on collected papers of the international conferences in the Natural Language Processing [14]

1. **Text dataset** for obtaining distributed representations
   - URLs were inserted into body texts

2. **Annotated URLs** for evaluating classification performances
   - we labeled 500 URLs appearing frequently in the collected papers
   - 100 URLs are development set

**Setup**
- Obtaining distributed representations: word2vec [17]
- For each method, the following parameter are selected based on the performance for the development set:
  - parameters of word2vec (and GRU)
  - classification model
  - whether to standardize input features

**Evaluation**
- 10 fold cross-validation for 400 annotated URLs
- metric
  - macro-averaged F1-score
  - F1-score for each label
### Experimental Result (1/2)

<table>
<thead>
<tr>
<th>Method</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>macro-ave</td>
</tr>
<tr>
<td>baseline (regarding each URL as a word)</td>
<td>0.779</td>
</tr>
<tr>
<td>summation</td>
<td>0.808</td>
</tr>
<tr>
<td>summation weighted by entropy</td>
<td>0.805</td>
</tr>
<tr>
<td>summation except top-level domains</td>
<td>0.816</td>
</tr>
<tr>
<td>GRU</td>
<td>0.820</td>
</tr>
<tr>
<td></td>
<td>tool</td>
</tr>
<tr>
<td>baseline (regarding each URL as a word)</td>
<td>0.830</td>
</tr>
<tr>
<td>summation</td>
<td>0.809</td>
</tr>
<tr>
<td>summation weighted by entropy</td>
<td>0.810</td>
</tr>
<tr>
<td>summation except top-level domains</td>
<td>0.821</td>
</tr>
<tr>
<td>GRU</td>
<td>0.835</td>
</tr>
<tr>
<td></td>
<td>data</td>
</tr>
<tr>
<td>baseline (regarding each URL as a word)</td>
<td>0.801</td>
</tr>
<tr>
<td>summation</td>
<td>0.725</td>
</tr>
<tr>
<td>summation weighted by entropy</td>
<td>0.732</td>
</tr>
<tr>
<td>summation except top-level domains</td>
<td>0.745</td>
</tr>
<tr>
<td>GRU</td>
<td>0.746</td>
</tr>
<tr>
<td></td>
<td>other</td>
</tr>
<tr>
<td>baseline (regarding each URL as a word)</td>
<td>0.663</td>
</tr>
<tr>
<td>summation</td>
<td>0.857</td>
</tr>
<tr>
<td>summation weighted by entropy</td>
<td>0.842</td>
</tr>
<tr>
<td>summation except top-level domains</td>
<td>0.864</td>
</tr>
<tr>
<td>GRU</td>
<td>0.865</td>
</tr>
</tbody>
</table>

- Obtaining distributed representations is effective for this task as a whole.
- baseline vs our approach
  - our approach got better results on macro-averaged F1 consistently
  - our approach was not good at discriminating the “data”
Experimental Result (2/2)

<table>
<thead>
<tr>
<th>Method</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>macro-ave</td>
</tr>
<tr>
<td>baseline (regarding each URL as a word)</td>
<td>0.779</td>
</tr>
<tr>
<td>summation</td>
<td>0.808</td>
</tr>
<tr>
<td>summation weighted by entropy</td>
<td>0.805</td>
</tr>
<tr>
<td>summation except top-level domains</td>
<td>0.816</td>
</tr>
<tr>
<td>GRU</td>
<td>0.820</td>
</tr>
</tbody>
</table>

- Comparing Compositional functions
  - Compared to the summation, weighting by entropy got worse results on some metrics
  - Compared to the summation, excluding top-level domains got better results on all metrics
  - GRU got the best results

there are useful URL elements in frequent URL elements and we should exclude top-level domains only
Conclusion & Future Work

• Conclusion
  – We formulate the URL classification task to realize the following things:
    ▪ identification of URLs citing research artifacts in scholarly papers
    ▪ generating information about the type of the research artifacts
  – Using distributed representations of URLs was effective, and using those of URL elements got better results
  – When synthesizing distributed representations of URL elements, excluding top-level domains is effective

• Future Work
  – reveal why our approach is not good at discriminating the “data”
  – more complex functions (e.g., using Transfer Encoder)
  – multi-label classification
    ▪ there are URLs distributing tools and datasets simultaneously
References (1/2)


[16] Aizawa Laboratory. PDFNLT 1.0. https://github.com/KMCS-NII/PDFNLT-1.0

References (2/2)


Thank you for listening!

Masaya Tsunokake (Presenter), Shigeki Matsubara
Nagoya University, Japan