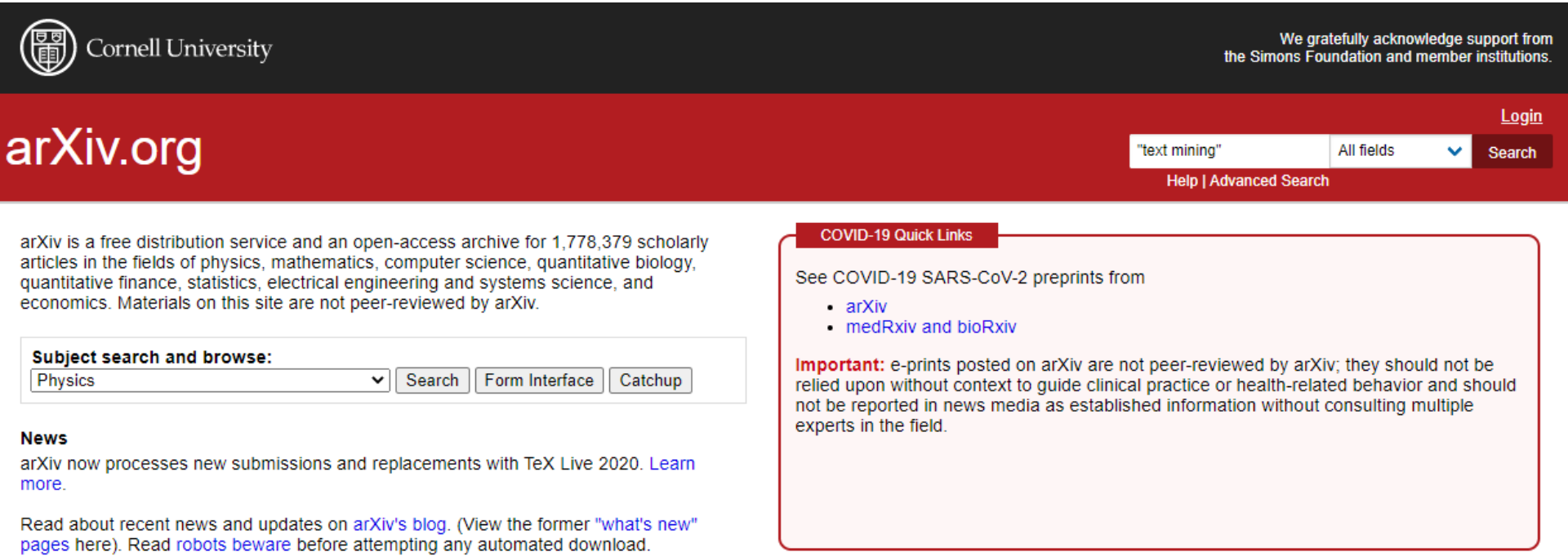
Web Scraping: arXiv Papers

Pilsung Kang
School of Industrial Management Engineering
Korea University

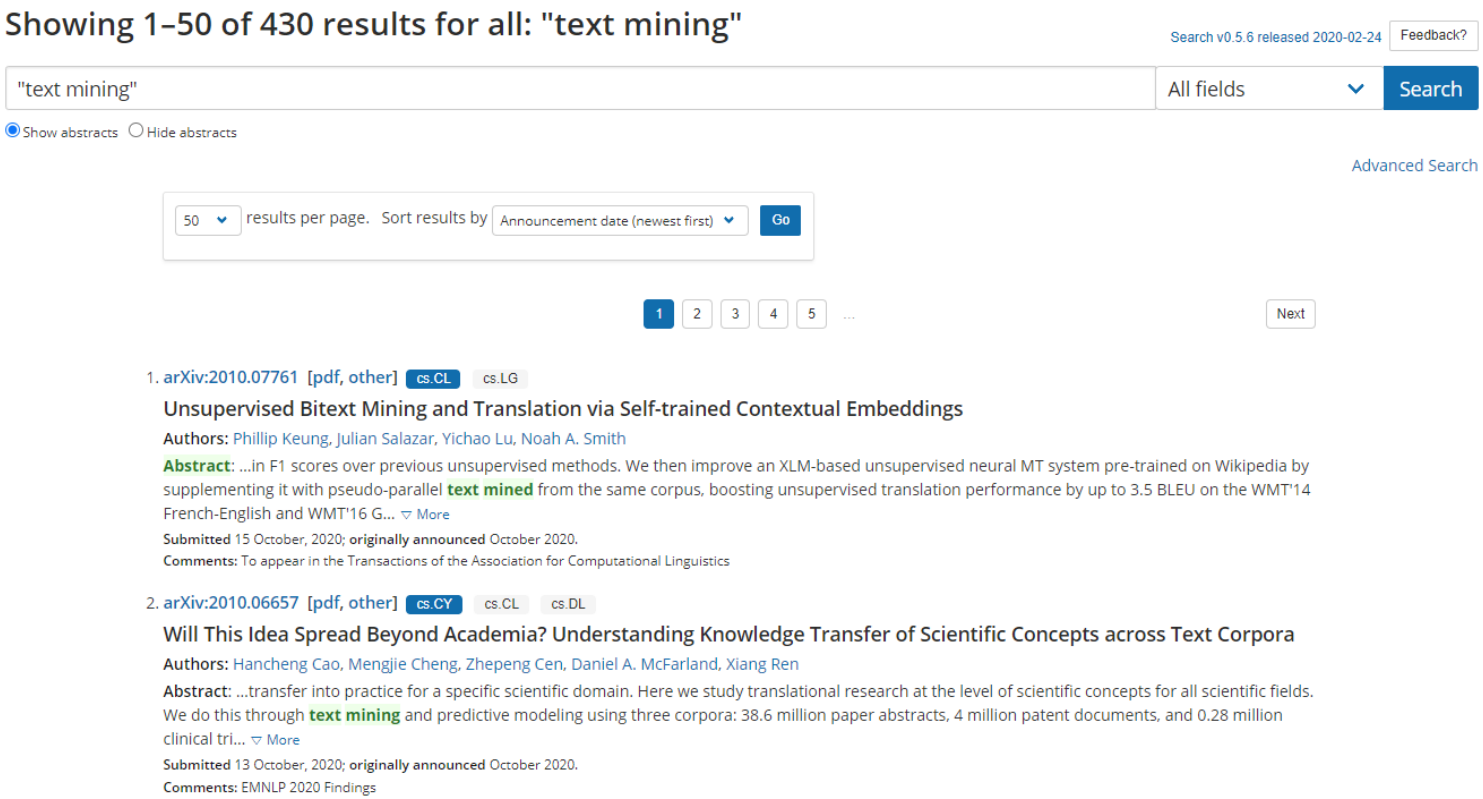
Web Scraping: arXiv Papers

- Web scraping example 1: arXiv papers about "Text Mining"
  - ✓ arXiv website: http://arxiv.org/
  - ✓ Collect Title, Authors, Subjects, Abstracts, and Meta Information



Web Scraping: arXiv Papers

- Step 1: Understand the basic structure
  - ✓ A total of 430 papers (2020-10-16), each page contains 50 papers (332 papers on 2019-10-07)
  - ✓ Each paper has a unique ID



Web Scraping: arXiv Papers

- Step 2: Analyzing the HTML Structure
  - ✓ First page URL
    - https://arxiv.org/search/?query=%22text+mining%22&searchtype=all&source=header&start=0
  - ✓ Second page URL
    - https://arxiv.org/search/?query=%22text+mining%22&searchtype=all&source=header&start=50
  - ✓ Third page URL
    - https://arxiv.org/search/?query=%22text+mining%22&searchtype=all&source=header&start=100

Web Scraping: arXiv Papers

- Step 2: Analyzing the HTML Structure
  - ✓ URL Parsing

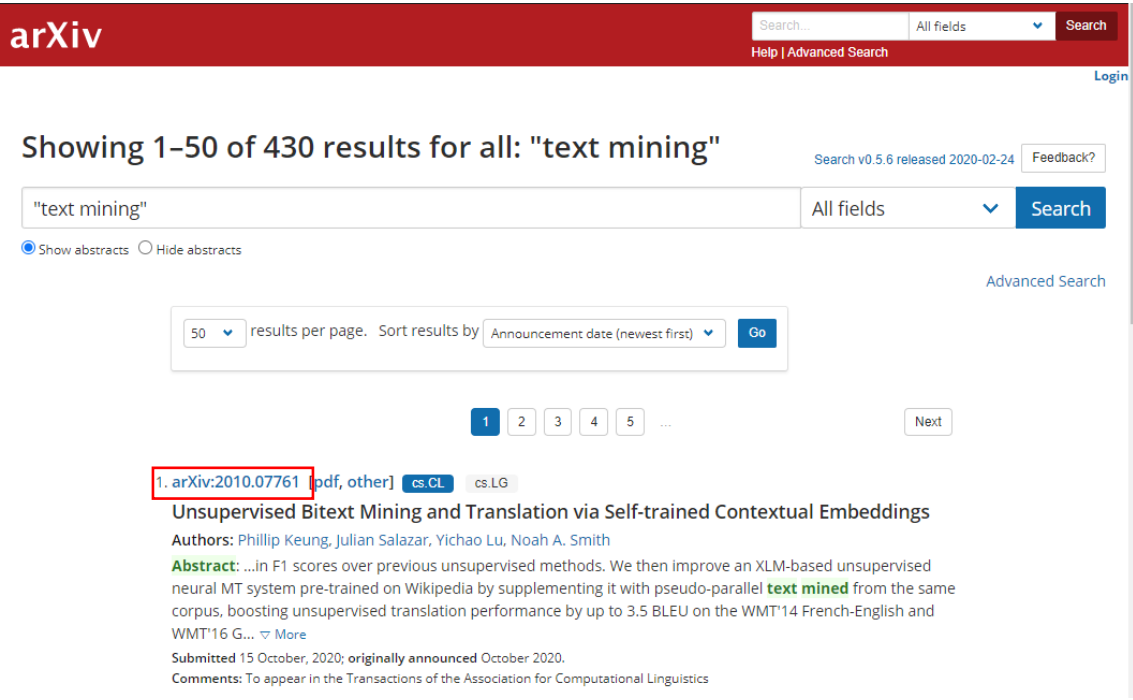
```
> parse_url(url)
$scheme
[1] "https"

$hostname
[1] "arxiv.org"

$port
NULL

$path
[1] "search/"

$query
$query$query
[1] "\"text+mining\""

$query$searchtype
[1] "all"

$query$source
[1] "header"

$query$start
[1] "0"

$params
NULL

$fragment
NULL

$username
NULL

$password
NULL

attr(,"class")
[1] "url"
```

The only part which actually changes

```
tmp_url <- modify_url(url, query = list(start = i))
```

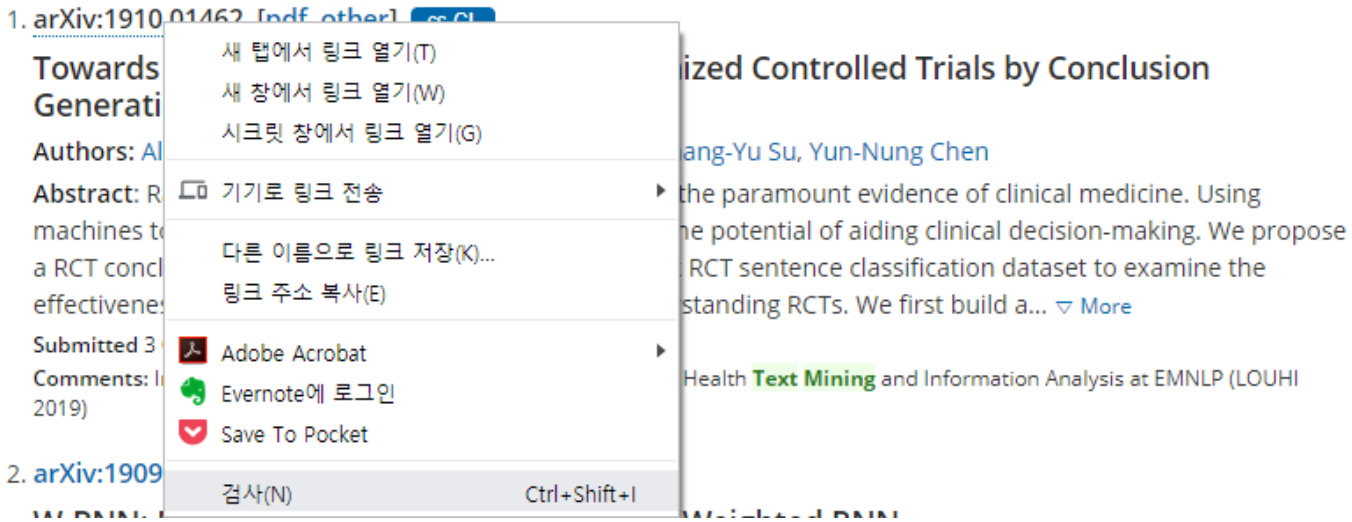
Web Scraping: arXiv Papers

- Step 2: Analyzing the HTML Structure (Press F12 in Chrome browser)
  - ✓ Find the node that contains the necessary links



Web Scraping: arXiv Papers

- Step 2: Analyzing the HTML Structure (Mouse right click ->)
  - ✓ Find the node that contains the necessary links




```
▼<div class="is-marginless" style="user-select: auto;">
  ▼<p class="list-title is-inline-block" style="user-select: auto;">
      <a href="https://arxiv.org/abs/2010.07761" style="user-select: auto;">
      arXiv:2010.07761</a> == $0
    ▶<span style="user-select: auto;">…</span>
    </p>
  ▶<div class="tags is-inline-block" style="user-select: auto;">…</div>
  </div>
```

Web Scraping: arXiv Papers

- Step 2: Analyzing the HTML Structure
  - ✓ Extract the link information
  - ✓ Should be familiar to the usage of CSS Selector
    - http://www.w3schools.com/cssref/css_selectors.asp

CSS Selectors

In CSS, selectors are patterns used to select the element(s) you want to style.

Use our CSS Selector Tester to demonstrate the different selectors.

The "CSS" column indicates in which CSS version the property is defined (CSS1, CSS2, or CSS3).

Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
element	p	Selects all <p> elements	1
element,element	div, p	Selects all elements and all <p> elements	1
element element	div p	Selects all <p> elements inside elements	1
element>element	div > p	Selects all <p> elements where the parent is a element	2
element+element	div + p	Selects all <p> elements that are placed immediately after elements	2
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element	3
[attribute]	[target]	Selects all elements with a target attribute	2
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"	2
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"	2
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"	2
[attribute^=value]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"	3
[attribute$=value]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"	3
[attribute*=value]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"	3

Web Scraping: arXiv Papers

- Step 2: Analyzing the HTML Structure
  - ✓ Extract the link information

```
tmp_list <- read_html(tmp_url) %>%
              html_nodes('p.list-title.is-inline-block') %>%
              html_nodes('a[href^="https://arxiv.org/abs"]') %>%
              html_attr('href')
```

  - find the node (p class = "list-title is –inline-block") → find the node whose href attribute begins with https://arxiv.org/abs→ Store the attribute value of 'href' to the tmp_list

  - ✓ Values that are stored in the "tmp_list"

```
> tmp_list
 [1] "https://arxiv.org/abs/2010.07761" "https://arxiv.org/abs/2010.06657" "https://arxiv.org/abs/2010.05194"
 [4] "https://arxiv.org/abs/2010.00732" "https://arxiv.org/abs/2010.00462" "https://arxiv.org/abs/2009.14797"
 [7] "https://arxiv.org/abs/2009.09223" "https://arxiv.org/abs/2009.08478" "https://arxiv.org/abs/2009.07642"
[10] "https://arxiv.org/abs/2009.07397" "https://arxiv.org/abs/2009.06376" "https://arxiv.org/abs/2009.05619"
[13] "https://arxiv.org/abs/2009.03087" "https://arxiv.org/abs/2008.12672" "https://arxiv.org/abs/2008.12277"
[16] "https://arxiv.org/abs/2008.10813" "https://arxiv.org/abs/2008.10749" "https://arxiv.org/abs/2008.07366"
[19] "https://arxiv.org/abs/2008.07343" "https://arxiv.org/abs/2008.07189" "https://arxiv.org/abs/2008.03911"
[22] "https://arxiv.org/abs/2008.03711" "https://arxiv.org/abs/2008.01937" "https://arxiv.org/abs/2007.12569"
[25] "https://arxiv.org/abs/2007.11053" "https://arxiv.org/abs/2007.06118" "https://arxiv.org/abs/2007.05651"
[28] "https://arxiv.org/abs/2007.04626" "https://arxiv.org/abs/2007.04100" "https://arxiv.org/abs/2007.03106"
[31] "https://arxiv.org/abs/2007.02237" "https://arxiv.org/abs/2007.00927" "https://arxiv.org/abs/2006.16642"
[34] "https://arxiv.org/abs/2006.15830" "https://arxiv.org/abs/2006.15311" "https://arxiv.org/abs/2006.11109"
[37] "https://arxiv.org/abs/2006.10315" "https://arxiv.org/abs/2006.06177" "https://arxiv.org/abs/2006.04042"
[40] "https://arxiv.org/abs/2006.00110" "https://arxiv.org/abs/2005.14080" "https://arxiv.org/abs/2005.11487"
[43] "https://arxiv.org/abs/2005.10595" "https://arxiv.org/abs/2005.09941" "https://arxiv.org/abs/2005.07465"
[46] "https://arxiv.org/abs/2005.07202" "https://arxiv.org/abs/2005.06889" "https://arxiv.org/abs/2005.06517"
[49] "https://arxiv.org/abs/2005.02799" "https://arxiv.org/abs/2005.00239"
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-1: Extract Title

Unsupervised Bitext Mining and Translation via Self-trained Contextual Embeddings

Phillip Keung, Julian Salazar, Yichao Lu, Noah A. Smith

We describe an unsupervised method to create pseudo-parallel corpora for machine translation (MT) from unaligned text. We use multilingual BERT to create source and target sentence embeddings for nearest-neighbor search and adapt the model via self-training. We validate our technique by extracting parallel sentence pairs on the BUCC 2017 bitext mining task and observe up to a 24.5 point increase (absolute) in F1 scores over previous unsupervised methods. We then improve an XLM-based unsupervised neural MT system pre-trained on Wikipedia by supplementing it with pseudo-parallel text mined from the same corpus, boosting unsupervised translation performance by up to 3.5 BLEU on the WMT'14 French-English and WMT'16 German-English tasks and outperforming the previous state-of-the-art. Finally, we enrich the IWSLT'15 English-Vietnamese corpus with pseudo-parallel Wikipedia sentence pairs, yielding a 1.2 BLEU improvement on the low-resource MT task. We demonstrate that unsupervised bitext mining is an effective way of augmenting MT datasets and complements existing techniques like initializing with pre-trained contextual embeddings.

Comments: To appear in the Transactions of the Association for Computational Linguistics
Subjects: **Computation and Language (cs.CL)**; Machine Learning (cs.LG)
Cite as: arXiv:2010.07761 **[cs.CL]**
(or arXiv:2010.07761v1 **[cs.CL]** for this version)

Submission history

From: Phillip Keung [view email]
[v1] Thu, 15 Oct 2020 14:04:03 UTC (4,478 KB)

```
▼<h1 class="title mathjax" style="user-select: auto;"> == $0
  <span class="descriptor" style="user-select: auto;">Title:</span>
  "Unsupervised Bitext Mining and Translation via Self-trained Contextual
  Embeddings"
</h1>
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-1: Extract Title

```
# title
tmp_title <- tmp_paragraph %>% html_nodes('h1.title.mathjax') %>% html_text(T)
tmp_title <- gsub('Title:', '', tmp_title)
title <- c(title, tmp_title)
```

- From tmp_paragraph → find the node whose h1 class name is "title mathjax" → extract the html text and store in to tmp_title

```
> tmp_title
[1] "Title:Unsupervised Bitext Mining and Translation via Self-trained Contextual Embeddings"
```

- Remove "Title:" from the tmp_title

```
> tmp_title
[1] "Unsupervised Bitext Mining and Translation via Self-trained Contextual Embeddings"
```

- Append the tmp_title to title

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-2: Extract Authors

Unsupervised Bitext Mining and Translation via Self-trained Contextual Embeddings

Phillip Keung, Julian Salazar, Yichao Lu, Noah A. Smith

We describe an unsupervised method to create pseudo-parallel corpora for machine translation (MT) from unaligned text. We use multilingual BERT to create source and target sentence embeddings for nearest-neighbor search and adapt the model via self-training. We validate our technique by extracting parallel sentence pairs on the BUCC 2017 bitext mining task and observe up to a 24.5 point increase (absolute) in F1 scores over previous unsupervised methods. We then improve an XLM-based unsupervised neural MT system pre-trained on Wikipedia by supplementing it with pseudo-parallel text mined from the same corpus, boosting unsupervised translation performance by up to 3.5 BLEU on the WMT'14 French-English and WMT'16 German-English tasks and outperforming the previous state-of-the-art. Finally, we enrich the IWSLT'15 English-Vietnamese corpus with pseudo-parallel Wikipedia sentence pairs, yielding a 1.2 BLEU improvement on the low-resource MT task. We demonstrate that unsupervised bitext mining is an effective way of augmenting MT datasets and complements existing techniques like initializing with pre-trained contextual embeddings.

Comments: To appear in the Transactions of the Association for Computational Linguistics
Subjects: **Computation and Language (cs.CL)**; Machine Learning (cs.LG)
Cite as: arXiv:2010.07761 **[cs.CL]**
(or arXiv:2010.07761v1 **[cs.CL]** for this version)

Submission history

From: Phillip Keung [view email]
[v1] Thu, 15 Oct 2020 14:04:03 UTC (4,478 KB)

```
▼<div class="authors" style="user-select: auto;">
  <span class="descriptor" style="user-select: auto;">Authors:</span>
  <a href="https://arxiv.org/search/cs?searchtype=author&query=Keung%2C+P"
  style="user-select: auto;">Phillip Keung</a> == $0
  ", "
  <a href="https://arxiv.org/search/cs?searchtype=author&query=Salazar%2C+J"
  style="user-select: auto;">Julian Salazar</a>
  ", "
  <a href="https://arxiv.org/search/cs?searchtype=author&query=Lu%2C+Y" style=
  "user-select: auto;">Yichao Lu</a>
  ", "
  <a href="https://arxiv.org/search/cs?searchtype=author&query=Smith%2C+N+A"
  style="user-select: auto;">Noah A. Smith</a>
</div>
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-2: Extract Authors

```
# author
tmp_author <- tmp_paragraph %>% html_nodes('div.authors') %>% html_text
tmp_author <- gsub('\\s+',' ',tmp_author)
tmp_author <- gsub('Authors:','',tmp_author) %>% str_trim
author <- c(author, tmp_author)
```

- From tmp_paragraph → Select node whose div class = "authors" → Store the html text
- Replace various spaces (space, tab, etc.) by a single space
- Remove 'Authors:" and trim the string

```
> tmp_author
[1] "Phillip Keung, Julian Salazar, Yichao Lu, Noah A. Smith"
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-3: Extract Subjects

Unsupervised Bitext Mining and Translation via Self-trained Contextual Embeddings

Phillip Keung, Julian Salazar, Yichao Lu, Noah A. Smith

We describe an unsupervised method to create pseudo-parallel corpora for machine translation (MT) from unaligned text. We use multilingual BERT to create source and target sentence embeddings for nearest-neighbor search and adapt the model via self-training. We validate our technique by extracting parallel sentence pairs on the BUCC 2017 bitext mining task and observe up to a 24.5 point increase (absolute) in F1 scores over previous unsupervised methods. We then improve an XLM-based unsupervised neural MT system pre-trained on Wikipedia by supplementing it with pseudo-parallel text mined from the same corpus, boosting unsupervised translation performance by up to 3.5 BLEU on the WMT'14 French-English and WMT'16 German-English tasks and outperforming the previous state-of-the-art. Finally, we enrich the IWSLT'15 English-Vietnamese corpus with pseudo-parallel Wikipedia sentence pairs, yielding a 1.2 BLEU improvement on the low-resource MT task. We demonstrate that unsupervised bitext mining is an effective way of augmenting MT datasets and complements existing techniques like initializing with pre-trained contextual embeddings.

Comments: To appear in the Transactions of the Association for Computational Linguistics
Subjects: **Computation and Language (cs.CL)**; Machine Learning (cs.LG)
Cite as: arXiv:2010.07761 **[cs.CL]**
(or arXiv:2010.07761v1 **[cs.CL]** for this version)

Submission history

From: Phillip Keung [view email]
[v1] Thu, 15 Oct 2020 14:04:03 UTC (4,478 KB)

```
▼<td class="tablecell subjects" style="user-select: auto;">
    <span class="primary-subject" style="user-select: auto;">Computation
    and Language (cs.CL)</span> == $0
    "; Machine Learning (cs.LG)"
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-3: Extract Subjects

```
# subject
tmp_subject <- tmp_paragraph %>% html_nodes('span.primary-subject') %>% html_text(T)
subject <- c(subject, tmp_subject)
```

- From tmp_paragraph → find the node whose span class = "primary-subject" → store the html text to tmp_subject

```
> tmp_subject
[1] "Computation and Language (cs.CL)"
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-4: Extract Abstract

Unsupervised Bitext Mining and Translation via Self-trained Contextual Embeddings

Phillip Keung, Julian Salazar, Yichao Lu, Noah A. Smith

We describe an unsupervised method to create pseudo-parallel corpora for machine translation (MT) from unaligned text. We use multilingual BERT to create source and target sentence embeddings for nearest-neighbor search and adapt the model via self-training. We validate our technique by extracting parallel sentence pairs on the BUCC 2017 bitext mining task and observe up to a 24.5 point increase (absolute) in F1 scores over previous unsupervised methods. We then improve an XLM-based unsupervised neural MT system pre-trained on Wikipedia by supplementing it with pseudo-parallel text mined from the same corpus, boosting unsupervised translation performance by up to 3.5 BLEU on the WMT'14 French-English and WMT'16 German-English tasks and outperforming the previous state-of-the-art. Finally, we enrich the IWSLT'15 English-Vietnamese corpus with pseudo-parallel Wikipedia sentence pairs, yielding a 1.2 BLEU improvement on the low-resource MT task. We demonstrate that unsupervised bitext mining is an effective way of augmenting MT datasets and complements existing techniques like initializing with pre-trained contextual embeddings.

Comments: To appear in the Transactions of the Association for Computational Linguistics
Subjects: **Computation and Language (cs.CL)**; Machine Learning (cs.LG)
Cite as: arXiv:2010.07761 **[cs.CL]**
(or arXiv:2010.07761v1 **[cs.CL]** for this version)

Submission history

From: Phillip Keung [view email]
[v1] Thu, 15 Oct 2020 14:04:03 UTC (4,478 KB)

```
▼<blockquote class="abstract mathjax" style="user-select: auto;"> == $0
  <span class="descriptor" style="user-select: auto;">Abstract:</span>
  "  We describe an unsupervised method to create pseudo-parallel corpora for
  machine translation (MT) from unaligned text. We use multilingual BERT to
  create source and target sentence embeddings for nearest-neighbor search and
  adapt the model via self-training. We validate our technique by extracting
  parallel sentence pairs on the BUCC 2017 bitext mining task and observe up to
  a
  24.5 point increase (absolute) in F1 scores over previous unsupervised
  methods.
  We then improve an XLM-based unsupervised neural MT system pre-trained on
  Wikipedia by supplementing it with pseudo-parallel text mined from the same
  corpus, boosting unsupervised translation performance by up to 3.5 BLEU on the
  WMT'14 French-English and WMT'16 German-English tasks and outperforming the
  previous state-of-the-art. Finally, we enrich the IWSLT'15 English-Vietnamese
  corpus with pseudo-parallel Wikipedia sentence pairs, yielding a 1.2 BLEU
  improvement on the low-resource MT task. We demonstrate that unsupervised
  bitext mining is an effective way of augmenting MT datasets and complements
  existing techniques like initializing with pre-trained contextual embeddings.

  "
</blockquote>
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-4: Extract Abstract

```
# abstract
tmp_abstract <- tmp_paragraph %>% html_nodes('blockquote.abstract.mathjax') %>% html_text(T)
tmp_abstract <- gsub('\\s+',' ',tmp_abstract)
tmp_abstract <- sub('Abstract:','',tmp_abstract) %>% str_trim
abstract <- c(abstract, tmp_abstract)
```

- From tmp_paragraph → find the node whose blockquote class = "abstract mathjax" → Store the html text to tmp_abstract
- Remove "Abstract:" and trim the text

```
> tmp_abstract
[1] "We describe an unsupervised method to create pseudo-parallel corpora for machine translation (M
T) from unaligned text. We use multilingual BERT to create source and target sentence embeddings for
 nearest-neighbor search and adapt the model via self-training. We validate our technique by extracti
ng parallel sentence pairs on the BUCC 2017 bitext mining task and observe up to a 24.5 point increas
e (absolute) in F1 scores over previous unsupervised methods. We then improve an XLM-based unsupervis
ed neural MT system pre-trained on Wikipedia by supplementing it with pseudo-parallel text mined from
 the same corpus, boosting unsupervised translation performance by up to 3.5 BLEU on the WMT'14 Frenc
h-English and WMT'16 German-English tasks and outperforming the previous state-of-the-art. Finally, w
e enrich the IWSLT'15 English-Vietnamese corpus with pseudo-parallel Wikipedia sentence pairs, yieldi
ng a 1.2 BLEU improvement on the low-resource MT task. We demonstrate that unsupervised bitext mining
 is an effective way of augmenting MT datasets and complements existing techniques like initializing
 with pre-trained contextual embeddings."
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-5: Extract Meta information

Unsupervised Bitext Mining and Translation via Self-trained Contextual Embeddings

Phillip Keung, Julian Salazar, Yichao Lu, Noah A. Smith

We describe an unsupervised method to create pseudo-parallel corpora for machine translation (MT) from unaligned text. We use multilingual BERT to create source and target sentence embeddings for nearest-neighbor search and adapt the model via self-training. We validate our technique by extracting parallel sentence pairs on the BUCC 2017 bitext mining task and observe up to a 24.5 point increase (absolute) in F1 scores over previous unsupervised methods. We then improve an XLM-based unsupervised neural MT system pre-trained on Wikipedia by supplementing it with pseudo-parallel text mined from the same corpus, boosting unsupervised translation performance by up to 3.5 BLEU on the WMT'14 French-English and WMT'16 German-English tasks and outperforming the previous state-of-the-art. Finally, we enrich the IWSLT'15 English-Vietnamese corpus with pseudo-parallel Wikipedia sentence pairs, yielding a 1.2 BLEU improvement on the low-resource MT task. We demonstrate that unsupervised bitext mining is an effective way of augmenting MT datasets and complements existing techniques like initializing with pre-trained contextual embeddings.

Comments: To appear in the Transactions of the Association for Computational Linguistics
Subjects: **Computation and Language (cs.CL)**; Machine Learning (cs.LG)
Cite as: arXiv:2010.07761 **[cs.CL]**
(or arXiv:2010.07761v1 **[cs.CL]** for this version)

Submission history
From: Phillip Keung [view email]
[v1] Thu, 15 Oct 2020 14:04:03 UTC (4,478 KB)

```
▼<div class="submission-history" style="user-select: auto;"> == $0
  <h2 style="user-select: auto;">Submission history</h2>
  " From: Phillip Keung ["
  <a href="/show-email/517c4aab/2010.07761" style="user-select: auto;">view email
  </a>
  "]
        "
  <br style="user-select: auto;">
  <strong style="user-select: auto;">[v1]</strong>
  "
  Thu, 15 Oct 2020 14:04:03 UTC (4,478 KB)"
  <br style="user-select: auto;">
</div>
```

Web Scraping: arXiv Papers

- Step 3: Extract necessary information
  - ✓ Step 3-5: Extract Meta information

```
# meta
tmp_meta <- tmp_paragraph %>% html_nodes('div.submission-history') %>% html_text
tmp_meta <- lapply(strsplit(gsub('\\s+', ' ',tmp_meta), '[v1]', fixed = T),'[',2) %>%
unlist %>% str_trim
meta <- c(meta, tmp_meta)
```

- From tmp_paragraph → find the node whose div class name is "submission-history" → Store the html text to tmp_meta

```
> tmp_meta
[1] "\n      Submission history From: Phillip Keung [view email]\n      [v1]\nThu, 15 Oct 2020 14:04:
03 UTC (4,478 KB)"
```

- Replace all spaces by a single space → Split the text (split point = [v1]) → Take the second element → Unlist it → trim the text

```
> tmp_meta
[1] "Thu, 15 Oct 2020 14:04:03 UTC (4,478 KB)"
```

Web Scraping: arXiv Papers

- Step 4: Repeat the process and export the data
  - ✓ Elapsed time for data collection

```
> end - start # Total Elapsed Time
 사용자  시스템 elapsed
  10.50    0.61  876.78
```

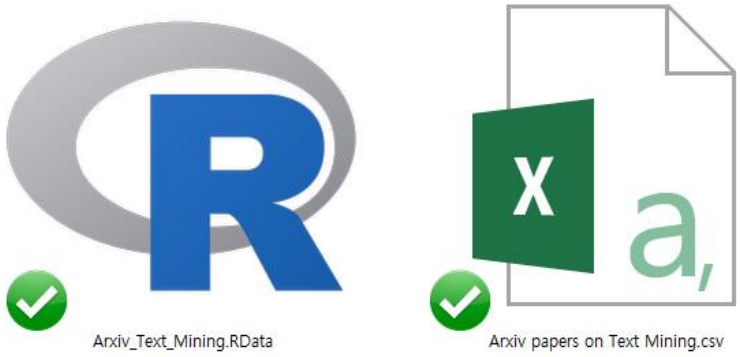
  - ✓ Check the dataset

Web Scraping: arXiv Papers

- Step 4: Repeat the process and export the data
  - ✓ Store the dataframe as an RData format or export it as a csv file

```
# Export the result
save(papers, file = "Arxiv_Text_Mining.RData")
write.csv(papers, file = "Arxiv papers on Text Mining.csv")
```

  - ✓ You can find the following two files in your working directory




ANY
questions?