GIANT: The 1-Billion Annotated Synthetic Bibliographic-Reference-String Dataset for Deep Citation Parsing

Mark Grennan[1 [0000−0001−9271−7444]], Martin Schibel[1 [0000−0003−1390−2874]], Andrew Collins[1 [0000−0002−0649−7391]], and Joeran Beel[1 [0000−0002−4537−5573]⋆]

Trinity College Dublin, School of Computer Science, ADAPT Centre, Ireland
grennama,ancollin,beelj @tcd.ie

Abstract. Extracting and parsing reference strings from research articles is a challenging task. State-of-the-art tools like GROBID apply rather simple machine learning models such as conditional random fields (CRF). Recent research has shown a high potential of deep-learning for reference string parsing. The challenge with deep learning is, however, that the training step requires enormous amounts of labelled data – which does not exist for reference string parsing. Creating such a large dataset manually, through human labor, seems hardly feasible. Therefore, we created GIANT. GIANT is a large dataset with 991,411,100 XML labeled reference strings. The strings were automatically created based on 677,000 entries from CrossRef, 1,500 citation styles in the citation-style language, and the citation processor citeproc-js. GIANT can be used to train machine learning models, particularly deep learning models, for citation parsing. While we have not yet tested GIANT for training such models, we hypothesise that the dataset will be able to significantly improve the accuracy of citation parsing. The dataset and code to create it, are freely available at https://github.com/BeelGroup/.

Keywords: Dataset · Deep Citation Parsing · Reference String Parsing · Document Engineering · Information Extraction

1 Introduction

Accurate citation parsing is important as citations are often used as a proxy for the strength of an academics career. In order to accurately report an researcher's citations, or accurately calculate an impact factor, journals, academic search engines and academic recommender systems must be able to extract citation metadata from each publication in their database. Failure to accurately parse citations could affect the validity of their results and consequently, an author's funding, status and future academic prospects.

⋆ This research was partially conducted at the ADAPT SFI Research Centre at Trinity College Dublin. The ADAPT SFI Centre is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106.

Citation parsing involves extracting machine readable metadata from a bibliography or citation string. As input, a citation parser accepts a citation string formatted in a particular citation style such as Harvard, APA, or IEEE. The parser then extract the metadata from the citation string and produces labelled output. The following citation string is formatted in Harvard style:

Councill, I.G., Giles, C.L. and Kan, M.Y., 2008, May. ParsCit: an Open-source CRF Reference String Parsing Package. In LREC (Vol. 8, pp. 661-667).

The corresponding labelled output is shown in Fig. 1. The output is typically formatted in XML with the field names included as XML tags. Here, the labelled output includes the authors' names, the date, the title of the article, the title of the journal, the volume, and the page numbers.

```
<bibl>
  <author>Councill, I.G.</author>
  <author>Giles, C.L.</author>
  <author>Kan, M.Y</author>
  <date>2008, May</date>
  <title>ParsCit: an Open-source CRF Reference String Parsing Package</title>
  <journal>LREC</LREC>
  <volume>8</volume>
  <pages>661-667</pages>
</bibl>
```

Fig. 1. An example of a citation string annotated in XML. Each field is encapsualted within its own tag.

In spite of it's importance citation parsing remains an open and difficult problem. In 2018 Tkaczyk et al. carried out a survey of ten open-source citation parsing tools, six machine-learning based tools and four non machine-learning based [16]. They reported that the ten tools had an average F1 of 0.56 and that ML-based tools outperformed non ML-based approaches by 133% (F1 0.77 for ML-based tools vs F1 0.33 for non-ML based tools). There remains room for significant improvement however a number of issues contribute to making this challenging. These include:

1. The large number of citation styles in use
2. The diversity of language used in citation strings
3. The diversity of citation types (e.g. books, journal articles, websites etc.)
4. The fact that each citation type may contain different fields and these fields are not known before processing
5. The presence of formatting errors
6. The lack of large amounts of labelled training data

The strength of a ML citation parser often reflects the quantity and quality of the training data. In order to train a ML citation parser to perform well on unseen citations each challenge must be addressed in the training data. Namely,

the training dataset should incorporate a diverse range of citation styles and citation types. It should contain citations from a broad range of disciplines and also some of the more common formatting errors. In order to satisfy all of these requirements the training dataset needs to be large.

Current training datasets for citation parsing have two fundamental problems. Firstly, they are homogeneous, with citations coming from a single domain. Many domains favour a particular citation style and training a model on only a few styles will not help it perform well across a range of domains. Further, limiting the training dataset to a single domain will reduce the diversity of domain-specific language the model is exposed to. Secondly, the majority of existing training datasets are small, having less than 8,000 labelled citations. It would be impossible for a training dataset of this size to fully reflect the diversity of citation styles or types that exist. Echoing these thoughts, a number of authors have commented on the potential benefits of having more training data available [4, 6].

With only small training datasets available common ML algorithms used for citation parsing include Support Vector Machines (SVM), Hidden Markov Models (HMM) and Conditional Random Fields (CRF). In 2018, Rodrigues et al. [15] and Prasad et al. [14] both separately applied a deep-learning approach to the problem of citation parsing. Although their training datasets are still relatively small - Rodrigues reported a training dataset of 40,000 citations - the results have been promising. Prasad et al. showed that Neural Parscit outperformed their earlier, non deep-learning approach.

Yet it remains to be seen what effect a much larger training dataset could have on the open problem of citation parsing. To fully leverage the potential of deep learning for citation string parsing, a large corpus is needed. Manually creating such a corpus does not seem reasonable and it is with this in mind that we introduce GIANT, a dataset of 1 billion annotated reference strings.

2 Related Work

2.1 Machine Learning Citation Parsers

We reviewed 31 papers on the topic of citation parsing and found that the number who adopt a ML approach to citation parsing greatly outnumbers the number that use non-ML methods (Fig. 2). Since 2010, 77% (24) of the 31 reviewed papers surveyed have adopted a ML-based approach. This perhaps reflects the growing consensus around the strengths of using ML methods. The four most common ML approaches are SVM, HMM, CRF and deep-learning. Fig. 3 shows the proportion of the 24 ML papers reviewed which used each model. Here, 12.5% used SVM, 29.2% used HMM, 50% used CRF and 8.2% used deep-learning.

Fig. 4 shows how the popularity of these ML models have changed over time. It highlights how HMM was more common pre-2010, CRF has remained consistently popular and a deep-learning approach has only been explored in the last two years.

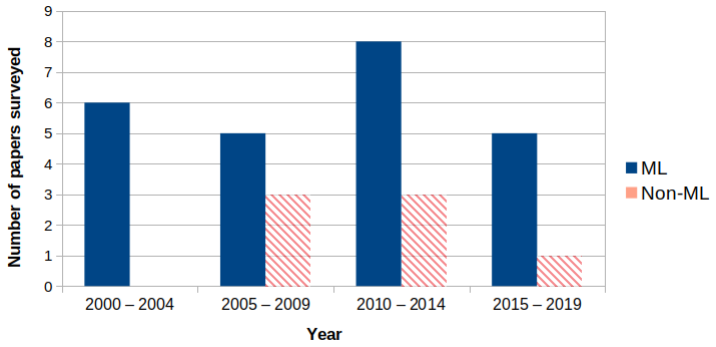

Fig. 2. The number of papers which have adopted a ML-based and a non ML-based approach to citation parsing between 2000 and 2019.

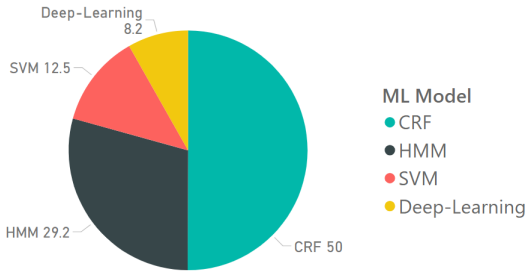

Fig. 3. The proportion of ML papers which used SVM, HMM, CRF and Deep-Learning

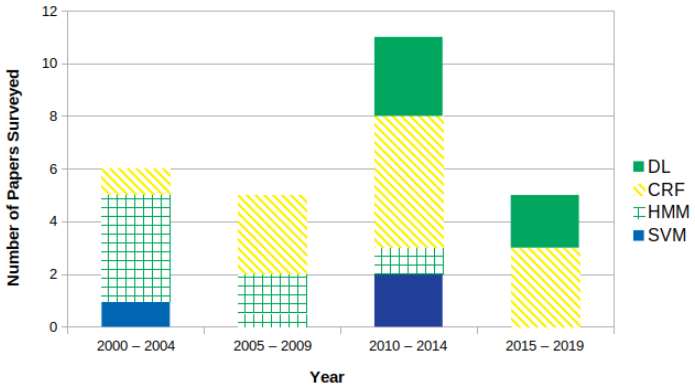

Fig. 4. The changing popularity of ML citation parsers between 2000 and 2019.

Tkaczyk et al. [16] showed that the performance of ML models can be improved by retraining a ML model on task-specific data. To improve a model's performance on a particular citation style, or domain-specific language, a ML model needs to be re-trained on relevant data.

2.2 Training Datasets

Table 1 summarises the size and domain of the training datasets used by eight ML citation parsing tools.

Table 1. Training datasets of eight ML citation parsing tools

Citation Parser	Training Dataset	Size	Domain
GROBID [9]	N/A	7800	N/A
Structural SVM [19]	PubMed	600	Health Science
HMM [7]	Cora	350	Computer Science
Bigram HMM [18]	ManCreat	712	N/A
Trigram HMM [12]	Cora + FluxCiM + ManCreat	1512	Computer Science
Deep Mining [15]	Venice	40000	Humanities
SVM + HMM [13]	Electronics, Communications & Computer Science	4651	Computer Science
CERMINE [17]	GROTOAP2	6858	Computer Science & Health Science

It is worth highlighting two points from this table. Firstly, many of these datasets were compiled from a single domain or sub-domain. Cora contains citations solely from Computer Science, PubMed contains citations from MEDLINE, a health science database and *Venice* contains citations from a corpus of documents on the history of Venice. As previously noted, many domains have their own domain-specific language and preferred citation style. Training a model on a single domain's technical language and only a few styles will not help it perform well across a range of domains.

The second point to note is the size of the training datasets. Aside from Rodrigues et al. [15] who have used a deep-learning approach and a training dataset of 40,000 citations, the remaining tools are trained on datasets smaller than 8,000 citations. Given the vast array of language and citation styles that exist it would be impossible for a training dataset of a such a size to fully capture this diversity. A number of authors have echoed these thoughts commenting on the limitations of existing datasets [4, 15, 14].

2.3 Deep Learning

Citation parsing can be defined as a sequence labelling problem. Advances have been made in recent years in the application of deep learning techniques

to sequence-labelling tasks. The state-of-the-art architectures for sequence labelling include a CRF prediction layer, word-embeddings and character-level word-embeddings. They are trained either with Convolutional Neural Networks (CNNs) [10] or Recursive Neural Networks (RNNs) using Bidirectional Long-Short Term Memory (BiLSTM) layers [8].

Rodrigues et al. [15] apply and compare the architectures of Lample et al. [8] and Ma and Hovy [10] to the task of reference mining. They define reference mining as the detection, extraction and classification of bibliographic references [15]. They trained and evaluated their model on citations extracted from a corpus of literature on the history of Venice. Word embeddings were pre-trained using Word2Vec [11] on the entire publications from which they extracted citations and the model was trained on 40,000 labelled citations. Extensive tuning was undertaken. Their final model outperformed a non deep-learning CRF baseline by 7.03% achieving an F1 of 0.896.

Prasad et al. [14] also examined how well a deep-learning approach would handle the task of citation parsing. Their final model deployed a Long Short-Term Memory (LSTM) neural network with a layered CRF over the output. In comparison against Parscit [5], their previous CRF-only based citation parser, they reported a significant ($p < 0.01$) improvement. However, they failed to show similar improvements in cross-domain performance and noted that a more diverse training dataset could help here.

Comparing the results of Prasad and Rodrigues is challenging. They both use a different CRF baseline and both models are trained and evaluated on different datasets. However, given that their available training data is relatively small their results are promising and highlight the potential of a deep-learning approach to the problem of citation parsing.

3 Goal and Methodology

Our goal was to create a large, diverse and annotated dataset that:

1. Is large enough to train deep neural networks
2. Contains a wide variety of citation styles
3. Contains many different citation types, e.g. journal article, books etc.
4. Contains a diverse range of language

3.1 Overview

GIANT was created using 677,000 bibliographic entries from Crossref. Each Crossref entry was reproduced in over 1,500 styles using Citation Style Languages (CSL) and the citation processor citeproc-js[2]. Reference management tools like Zotero, Mendeley or Docear make use of CSL to automatically generate references and bibliographies in any required citation style. CSL is an XML-based language that provides a machine-readable description of citation styles and bibliographic format. There are three main components that are used by a

CSL processor to generate a reference string. These are an item's citation style, metadata information and locale file. As shown in Fig. 5, combing these three items in a CSL processor will produce a citation string. In Fig. 5 the following citation string is produced: *M. Grennan, 1st August 2019, The 1 Billion Dataset.*

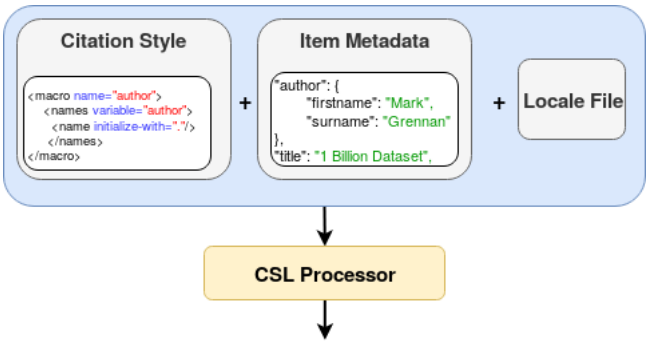

Fig. 5. Combining a CSL Style, an item's metadata and locale file in a CSL processor will produce a citation string.

In CSL a citation style is an XML file that defines the format of the citation string. Each referencing format - Harvard, IEEE etc. - has it's own citation style. An item's metadata stores the bibliographic details of the entry you wish to cite. This may include the author's names, the title, date etc. and common formats for storing an item's metadata include BibTeX, RIS and JSON. Finally, locale files are used to define language-specific phrases in a citation string. In creating GIANT we used the US-English locale file and all reference strings in GIANT are in the English language.

Training data for ML citation parsers must be labelled XML citation strings such as that shown in Fig. 6. In order to create this training data the XML citation styles were edited. These edited citation styles, along with a locale file and an items metadata, were then combined with a CSL processor to produce the desired labelled citation strings.

```
<author>M. Grennan</author>, <date>1st August 2019</date>,
<title>The 1 Billion Dataset</title>
```

Fig. 6. An example of labelled XML which is used as input to a train a ML citation parser.

Fig. 7 gives a high-level overview of the process. The citation styles were edited and combined with citation metadata records and a locale file in a CSL processor. The final output is labelled XML citation strings.

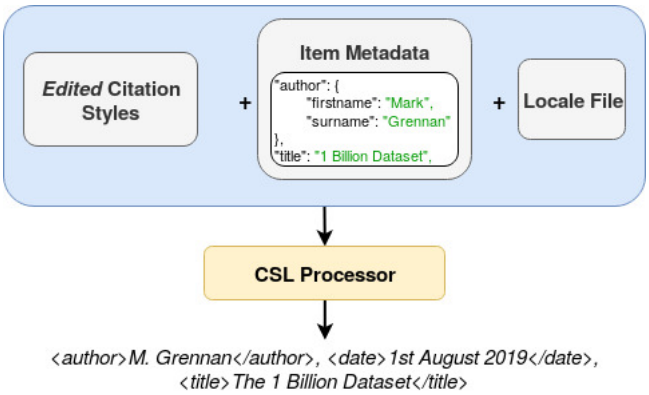

Fig. 7. An overview of the process of creating GIANT. Citation styles were edited and combined with citation metadata to produce a labelled citation string.

3.2 Editing Citation Styles

To make GIANT diverse, we included 1,564 different XML citation styles obtained from the official CSL repository [3]. The first task in creating GIANT was to edit these styles so that each field would contain a prefix tag (<author>, <title> etc.) and a suffix tag (</author>, </title> etc.). Table 3.2 gives examples of fields both before and after the prefix and suffix tags were added.

Table 2. Original CSL tags and CSL tags after a prefix and suffix tag has been added.

Field	Original CSL	Edited CSL
publisher	<text variable="publisher"/>	<text variable="publisher" prefix="<publisher>" suffix="</publisher>"/>
date	<date-part name="year"/>	<date-part name="year" prefix="<year>" suffix="</year>"/>

Different citation parsing tools require slightly different formatting for their training data. Some tools, such as GROBID [9], require that all author names are contained within a single author tag whilst other tools, such as Parscit [5], require individual authors to be encapsulated within their own tag. To make GIANT as

widely usable as possible an author's first name, middle name and surname were given separate tags. A *family* tag was used to represent the author's surname and a *given* tag was used to represent their first name and/or middle name.

Fig. 8 shows an example of a macro for author before and after tags for author, family and given are added. The prefix and suffix for *family* and *given* are contained within each individual name-part tag. Before editing, the macro in Fig. 8 will produce the following string: *M. Grennan*. After editing, the macro will produce the following labelled author field:

<author><given>M.</given> <family>Grennan</family></author>

Before Editing

```
<macro name="author">
  <names variable="author">
    <name initialize-with="."/>
  </names>
</macro>
```

After Editing

```
<macro name="author">
  <names variable="author" prefix="&lt;author>" suffix="&lt;/author>">
    <name initialize-with="."/>
      <name-part name="family" prefix="&lt;family>" suffix="&lt;/family>"/>
      <name-part name="given" prefix="&lt;given>" suffix="&lt;/given>"/>
    </name>
  </names>
</macro>
```

Fig. 8. The macro for author before and after the name-part tags have been added for the fields "family" and "given". Note, < is used to represent the special character < in XML.

Should a citation contain multiple authors or editors their names will be contained within the outer author tag, for example:

<author><given>M.</given> <family>Grennan</family>,
<given>U.</given> <family>McMenamin</family></author>

3.3 Item Metadata and Crossref

In order to obtain diversity in domain and citation type a large source of accessible citation metadata is required. One large, freely-available source of scholarly metadata is Crossref [1]. CrossRef is a not-for-profit organisation that collates, tags and shares metadata on scholarly publications. Their records contain over a hundred billion records from a diverse range of academic fields. 677,000 random records were obtained from Crossref using their public API and their *random_doi* method.

3.4 CSL Processor

Citeproc-js [2] was chosen as the CSL processor for the following reasons. It has been in operation for over a decade, it is open-source and it is widely used.

In order to use citeproc-js the input data must be in JSON form and follow the citeproc-js JSON schema. Crossref returned metadata in JSON form but a number of steps were required to make this JSON compatible with the citeproc-js JSON schema. These steps included changing tag names and removing any empty tags or tags not compatible with the citeproc-js JSON schema.

3.5 Indexes

In an effort to provide information for future users of GIANT three pieces of metadata were included with each labelled citation. These were:

1. The DOI of the citation
2. The citation type (book, journal article etc.)
3. The citation style used (Harvard, MLA etc.)

Both the citation style and the citation type were included as indexes and a separate lookup table is provided for both.

4 Results

The source code to create GIANT and instructions on how to download the dataset (438GB) are available on GitHub https://github.com/BeelGroup/. The final format of GIANT is a CSV file with four columns: DOI, citation type, citation style and labelled citation string. Table 3 gives an example of the layout.

Table 3. The final format of GIANT. Columns exist for DOI, citation type, citation style and XML labelled citation.

DOI	Type	Style	Labelled Citation String
10.1186/s12967-016-0804-1	3	471	<author><family>Yang</family> etc.
10.1037/ser0000151	3	1084	<author><family>Goetter</family> etc.

GIANT comprises of 633,895 unique reference strings, each available in 1,564 styles, resulting in a total of 991,411,100 labelled citation strings. Fig. 9 gives the percentage breakdown of GIANT by citation type. Journal articles are the most common type of citation making up 75.9% of GIANT, followed by chapter citations, 12.4% and conference papers, 5.6%.

Table 4 provides further detail with columns included for total number of labelled citations, number of unique citation strings and percentage of dataset.

Table 4. Breakdown of Citation Types contained within GIANT

Citation Type	Labelled Citation Strings	Unique Citation Strings	Percentage
Journal Article	752,005,608	480,822	75.9%
Chapter	122,562,727	78,365	12.4%
Conference Paper	55,706,868	35,618	5.6%
Dataset	17,003,027	10,872	1.7%
Reference Entry	8,371,603	5,353	0.8%
Book	7,077,100	4,525	0.7%
Other	28,684,167	18,340	2.9%
Total	**991,411,100**	**633,895**	**100%**

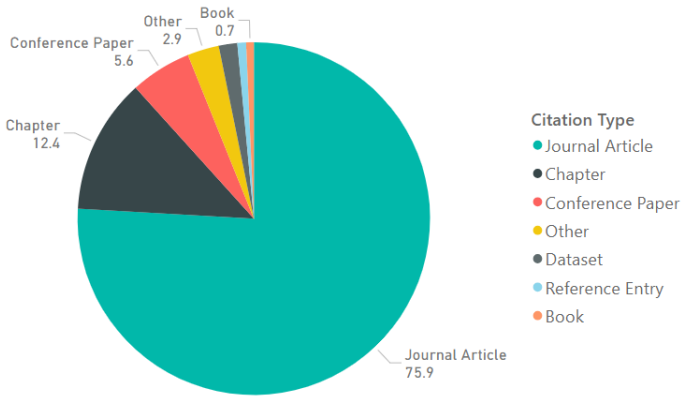

Fig. 9. The percentage breakdown of citation types contained within GIANT

5 Limitations and Future Work

In GIANT, a *container-title* tag is used interchangeably to represent journal titles, book titles and series titles. This is a potential disadvantage as some citation parsing tools use different tags for each of these items. For example, they may use a *journal* tag for a journal title and a *book* tag for a book title. This disadvantage can be overcome by using the citation type lookup index provided to map the *container-title* tag to more meaningful labels such as: journal, book, conference-paper etc.

As detailed in the related work the majority of existing citation parsing tools use small, hand-labelled training datasets. Many diverse fields have made significant advances in recent years due to the availability of more data and the application of deep-learning. The work of Rodrigues et al. [15] and Prasad et al. [14] in 2018 has given an early indication that citation parsing is also likely to benefit from applying deep-learning methods. The obvious area for future work would be to train a deep-learning citation parsing tool using GIANT.

GIANT is many orders of magnitude greater than any other available training dataset. It has been shown to be diverse in citation style, type and domain. Training a deep learning citation parsing tool with GIANT has the potential to significantly improve the accuracy of citation parsing.

References

1. Crossref, https://www.crossref.org
2. A JavaScript implementation of the Citation Style Language (CSL), https://github.com/Juris-M/citeproc-js
3. Official repository for Citation Style Language (CSL), https://github.com/citation-style-language/styles

4. Anzaroot, S., McCallum, A.: A New Dataset for Fine-Grained Citation Field Extraction (2013)
5. Councill, I.G., Giles, C.L., Kan, M.Y.: Parscit: an open-source crf reference string parsing package. In: LREC. vol. 8, pp. 661–667 (2008)
6. Fedoryszak, M., Tkaczyk, D., Bolikowski, Ł.: Large scale citation matching using apache hadoop. In: International Conference on Theory and Practice of Digital Libraries. pp. 362–365. Springer (2013)
7. Hetzner, E.: A simple method for citation metadata extraction using hidden markov models. In: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries. pp. 280–284. ACM (2008)
8. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
9. Lopez, P.: Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In: International conference on theory and practice of digital libraries. pp. 473–474. Springer (2009)
10. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
12. Ojokoh, B., Zhang, M., Tang, J.: A trigram hidden markov model for metadata extraction from heterogeneous references. Information Sciences **181**(9), 1538–1551 (2011)
13. Okada, T., Takasu, A., Adachi, J.: Bibliographic component extraction using support vector machines and hidden markov models. In: International Conference on Theory and Practice of Digital Libraries. pp. 501–512. Springer (2004)
14. Prasad, A., Kaur, M., Kan, M.Y.: Neural parscit: a deep learning-based reference string parser. International Journal on Digital Libraries **19**(4), 323–337 (2018)
15. Rodrigues Alves, D., Colavizza, G., Kaplan, F.: Deep reference mining from scholarly literature in the arts and humanities. Frontiers in Research Metrics and Analytics **3**, 21 (2018)
16. Tkaczyk, D., Collins, A., Sheridan, P., Beel, J.: Machine learning vs. rules and out-of-the-box vs. retrained: An evaluation of open-source bibliographic reference and citation parsers. In: Proceedings of the 18th ACM/IEEE on joint conference on digital libraries. pp. 99–108. ACM (2018)
17. Tkaczyk, D., Szostek, P., Dendek, P.J., Fedoryszak, M., Bolikowski, L.: Cermine–automatic extraction of metadata and references from scientific literature. In: 2014 11th IAPR International Workshop on Document Analysis Systems. pp. 217–221. IEEE (2014)
18. Yin, P., Zhang, M., Deng, Z., Yang, D.: Metadata extraction from bibliographies using bigram hmm. In: International Conference on Asian Digital Libraries. pp. 310–319. Springer (2004)
19. Zhang, X., Zou, J., Le, D.X., Thoma, G.R.: A structural svm approach for reference parsing. BMC bioinformatics **12**(3), S7 (2011)