1 Billion Citation Dataset and Deep Learning Citation Extraction

Mark Grennan

A dissertation submitted to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Future Networked Systems)

Supervisor: Joeran Beel

August, 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Mark Grennan
August 8, 2019

Permission to Lend/Copy

I, the undersigned, agree that Trinity College Library may lend or copy this dissertation upon request.

Mark Grennan
August 8, 2019

Acknowledgements

I would like to thank my supervisor Joeran Beel for his attention to detail, consideration and thoughtful advice throughout this project. I would also like to thank Martin Schibel for his prior work and for giving generously with his time in the early stages of the project. Finally, for their selfless patience and encouragement I would like to thank Una and my family.

Mark Grennan
August, 2019

1 Billion Citation Dataset and Deep Learning Citation Extraction

Mark Grennan, Master of Science in Computer Science
University of Dublin, Trinity College, 2019

Supervisor: Joeran Beel

Abstract

Citation or reference parsing involves extracting machine readable metadata from a citation string. This paper details the work carried out in creating a large, diverse and labelled citation dataset which can be used to train ML citation parsing tools. The dataset was created by adapting the citation styles within CSL, collecting citation metadata from CrossRef and using the open-source citation processor, citeproc-js. It contains 991,411,100 XML labelled citation strings in over 1,500 different citation styles. These are composed of a diverse range of citation types from a wide range of academic fields. When the granularity of the training data is the same the 1 Billion Citation Dataset was shown to be on a par with the state-of-the-art hand-labelled training datasets (Hand-labelled Training Data Macro-Average F1 0.74 vs Synthetic Training Data Macro-Average F1 0.74). These results indicate that the 1 Billion Citation Dataset has a lot of potential for training a deep-learning based citation parsing model and that this may lead to significant improvements in the accuracy of citation parsing.

Contents

List of Tables

List of Figures

Abbreviations

ML	Machine Learning
CRF	Conditional Random Fields
SVM	Support Vector Machines
HMM	Hidden Markov Models
CNN	Convolution Neural Networks
RNN	Recursive Neural Networks
XML	Extensible Markup Language
OCR	Optical Character Recognition
CSL	Citation Styles Language
CoNLL	Conference on Natural Language Learning
MLN	Markov Logic Networks
BiLSTM	Bidirectional Long-Short Term Memory
LSTM	Long Short-Term Memory
JSON	JavaScript Object Notation
RIS	Research Information Systems

Chapter 1

Introduction

1.1 Background

The accuracy of citation parsing is important for a number of reasons. The quantity and quality of citations is a commonly accepted proxy for the strength of an academic's career [3, 4, 5]. The total number of citations attributed to an academic can affect their career trajectory, whether they receive funding [6] and ultimately, their academic legacy [7].

In order to accurately report an author's citations search engines such as Scopus [8], Web of Science [9] and Google Scholar [10] 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 subsequently, an author's funding, status and future academic prospects.

Similarly, the impact factor of journal is a commonly accepted proxy for relative importance [11, 12]. The calculation of an impact factor relies on accurate information retrieval and citation parsing [13]. A journal's impact factor can affect individuals, journals, departments, universities [13] and even whole countries [14].

Finally, academic search engines such as Google Scholar [10], Refseek [15], and Microsoft Academic Search [16], and academic recommender systems, such as Mr Dlib [17], all rely on accurate citation metadata for data organisation and information retrieval [18]. The relevance of their search results relies on the ability to accurately parse citations.

Citation parsing involves extracting machine readable metadata from a publication, bibliography or citation string. A citation parser accepts a citation string as input. This string is typically formatted in a particular citation style: Harvard, APA, IEEE etc. The job of a citation parser is to extract the metadata from the given citation string and produce 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 Figure 1.1. This 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. Citation parsing can be seen as undoing the process of formatting a citation string [1].

```
<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>
```

Figure 1.1: An example of a citation string annotated in XML. Each field is encapsualted within it's own tag.

There are a number of existing approaches to citation parsing one of which is supervised machine-learning. A supervised machine-learning approach requires labelled training data, an algorithm and a labelled test dataset.

Popular Machine Learning algorithms used for citation parsing include Support Vector Machines [19, 20], Hidden Markov Models [21, 22, 23, 24] and Conditional Random Fields [18, 25, 26, 27, 28, 29, 30, 31]. Prasad et al. [13] and Rodrigues et al. [32] are the only published authors who have used a deep learning based approach. Tkaczyk et al. recently presented ParsRec, a recommender-system which suggests the best citation parser for a given citation string [33].

Finally, the strength of a citation parser is typically evaluated using the metrics: recall, precision and F1 with the evaluation often subdivided by field as shown in Table 1.1.

Field	Precision	Recall	F1
Author	0.94	0.94	0.94
Journal	0.16	0.70	0.26
Volume	0.76	0.19	0.31
Pages	0.91	0.58	0.71
Date	0.84	0.64	0.72
Average	0.72	0.61	0.59

Table 1.1: Sample evaluation metrics subdivided by field.

1.2 Research Problem

In spite of it's importance citation parsing remains an open and difficult problem. In 2018 Tkaczyk et al. [1] carried out a comparison survey of ten open-source citation parsing tools, six machine-learning based tools and four non machine-learning based. They reported that the ten tools had an average F1 of 0.56, 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) and that the highest three performing tools were GROBID (F1 0.89) [26], CERMINE (F1 0.83) [18] and Parscit (F1 0.75) [25].

There remains room for significant improvement in the field of citation parsing however a number of issues contribute to making this challenging.

1. Citation Styles
   A significant challenge associated with citation parsing is the existence of thousands of different citation styles [34] [25]. In formatting a citation string certain information may be removed or abbreviated depending on the particular citation style. Table 1.2 shows the same citation formatted in Harvard and ACM styles. In the Harvard style the authors' first and middle names are abbreviated to their initials and included after the authors' surname. This is followed by the year and month of publication. The terms *Vol.* and *pp.* are included to indicate volume and page number. In ACM style the authors' full names are included, there are brackets around the date and neither volume or page number are indicated by any proceeding terms. These differences are multiplied by the thousands of existing citation styles. The challenge in citation parsing is how to extract this information without prior knowledge of the citation style.

Style	Citation
Harvard	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).
ACM	Isaac G Councill, C Lee Giles, and Min-Yen Kan. ParsCit: An open-source CRF reference string parsing package. LREC 8 (May 2008), 661-667

Table 1.2: A citation string formatted in Harvard and ACM citation style. Each style has a different format for author, date, volume and page number.

2. Citation Types
   A second difficulty associated with citation parsing is that the type of citation is not known beforehand [1]. Citations for books, conference papers, journal articles, websites, blog posts etc. usually contain different information. A citation for a website would typically include a

field for the URL and date-accessed, whilst a citation for a book would often include a field for the publisher and book-title. Without knowing the type of citation before parsing it is difficult to know what fields it should contain.

3. Language Diversity
   Another challenge for citation parsing is that the model must perform well on a citations from a broad range of disciplines. Many disciplines contain domain-specific language. For example, the vocabulary of Microbiology may have little in common with Astrophysics. Capturing this diversity remains a challenge.

4. Errors
   Finally, it is common for citations to contain formatting errors. Manually formatting citations can introduce human-errors such as: missing or extra spaces, typos, missing punctuation and style-specific errors [1] [35]. Optical Character Recognition (OCR) errors can also be introduced during the process of converting scanned images into electronic versions. Common OCR errors are: substitution errors, disambiguating between similar looking characters and breaking words into multiple pieces [36].

The strength of a Machine Learning citation parser often reflects the quantity and quality of the training data [35]. In order to train a Machine Learning citation parser to perform well on unseen citations each of the aforementioned challenges must be addressed in the training data. Namely, the training dataset should incorporate the 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. This is problematic as many domains favour a particular citation style. For example, ACM is a popular style in Computer Science whilst Modern Languages Association (MLA) is a popular style in the Humanities. 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 [37, 38] and the limitations of existing datasets [32].

Although training a Machine Learning citation parser on a large and diverse dataset may lead to improvements in performance, producing such a large, annotated dataset is not straightforward. Remember that the aim of citation parsing is to produce labelled data such as that shown in Figure 1.1 when given a citation string. However, a Machine Learning citation parser requires such labelled data for training. So there exists the frustrating cycle of requiring a citation parser to produce labelled training data, which can then subsequently be used to train a citation parser.

This goes a long way to explaining why the majority of existing datasets are relatively small and homogeneous. Each of these datasets would probably have had to be compiled and annotated manually, perhaps with the help of an existing citation parser, and this unenviable task would be very labour intensive [19].

With only small training datasets available the majority of ML algorithms used for citation parsing are relatively dated. These include SVM (1960s) [39], HMM (1960s) [40] and CRF (2001) [41]. A lot of recent advances in ML can be attributed to deep-learning however, in spite of artificial neural networks being around since McCulloch and Pitts in the 1940s [42], it was only in the 2000s, as more data became available, that deep-learning was able to overtake many traditional non deep-learning methods.

In 2018, Rodrigues et al. [32] and Prasad et al. [13] 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 [32] - the results have been promising. Prasad et al. showed that Neural Parscit outperformed their earlier, non deep-learning approach, Parscit, with statistical significance, reducing the macro error rate from 11.17% to 8.63% [13].

Yet it remains to be seen what effect a much larger training dataset could have on the open problem of citation parsing. In much the same way a large labelled dataset transformed the landscape for image classification, the presence of a freely available, large, diverse and labelled citation dataset may enable significant further advances to be made in the area of citation parsing through the application of deep learning.

1.3 Research Question

The long-term goal of this project is to address the question:

- How would training a deep-learning citation parsing tool on a large, diverse synthetic dataset affect the accuracy of citation parsing?

The scope of this project does not include training a deep-learning model so this dissertation will not give a conclusive answer to this question. However, in order to investigate the suitability of a large, synthetic dataset for training a deep-learning model I will examine the following questions:

- How does a synthetic dataset compare to hand-labelled datasets when training existing ML citation parsering tools?
- How does the size of the training dataset affect the performance of ML citation parsers?
- How does the make-up of the training dataset (distribution of citation types etc.) affect the performance of ML citation parsers?

1.4 Research Aims

The requirements for a training dataset for ML citation parsing are that it is diverse and large. It should be diverse in citation styles, types and domain and, in order to fully reflect this diversity, it needs to be large.

As outlined in Section 1.2 current available training datasets are both homogeneous and small. In an effort to address this problem the primary aim of my research is to produce a large, diverse and labelled training dataset.

The overall goal is to then investigate whether such a large and diverse synthetic training dataset has the potential to be used for training a deep-learning citation parsing model. In order to address this, I aim to investigate how existing citation parsing tools compare when trained on both the synthetic dataset and the out-of-the-box hand-labelled datasets.

The final task will be to experiment with different dataset sizes and make-ups to determine if these make a significant difference to the performance of a ML citation parser.

1.5 Prior Work

It should be noted that some of the source code used in this project was previously developed by Martin Schibel, an intern working under Joeran Beel at the ADAPT centre.

Chapter 2

Background

2.1 Citation Strings, Styles and Fields

Some authors have made a distinction between the words "citation" and "reference" using one word to refer to items contained in the body of a document (e.g. Grennan 2019) and another to refer to items contained within a bibliography at the end of a document [25]. However, "citation parsing" and "reference parsing" are often used interchangeably. Here, a citation string, such as:

I. G. Councill, C. L. Giles, and M.-Y. Kan, "ParsCit: An open-source CRF reference string parsing package", LREC, vol. 8, pp. 661-667, May. 2008.

refers to an individual entry found in the bibliography at the end of a document.

As outlined in Section 1.2 a citation string is commonly formatted in a particular citation style: ACM, IEEE, Harvard etc. The number of unique citation styles is in the region of 1,000 [43] however, when variations of styles are included this number increases to approximately 8,000 [44]. Often academic fields or journals will have their own preference for a particular citation style.

The fields of a citation string may also vary. Common fields include author, title and date however the exact fields present in a citation will depend on the type of citation. The fields present in a website citation will typically be different to those present in a book citation. A summary of common citation types and their respective fields is given in Table 2.1. It is also worth noting that Table 2.1 is far from exhaustive and citations sometimes omit fields depending on the citation style and the available information.

Citation Type	Fields
Journal Article	Author, Date, Article Title, Journal Title, Volume, Issue, Page, DOI
Conference Paper	Author, Date, Title, Paper Title, Editor, Publisher, Publisher Location, DOI
Chapter	Author, Date, Chapter Title, Book Title, Editor, Publisher, Publisher Location, DOI
Book	Author, Date, Book Title, Editor, Publisher, Publisher Location, DOI
Website	Author, URL, Date Accessed
Report	Author, Date, Title, Editor, Publisher, Publisher Location, DOI

Table 2.1: Common citation types and their respective fields.

2.2 Citation Parsing

2.2.1 Introduction

Parsing a citation involves splitting a citation string into it's respective fields. Citation parsing often exists as part of a larger workflow of document processing. Figure 2.1 gives a typical example of the workflow of extracting citation metadata from a PDF document.

First, the bibliography section of the document is identified, second the bibliography is split into individual citation strings and finally, the metadata is retrieved from each individual citation string by a citation parser.

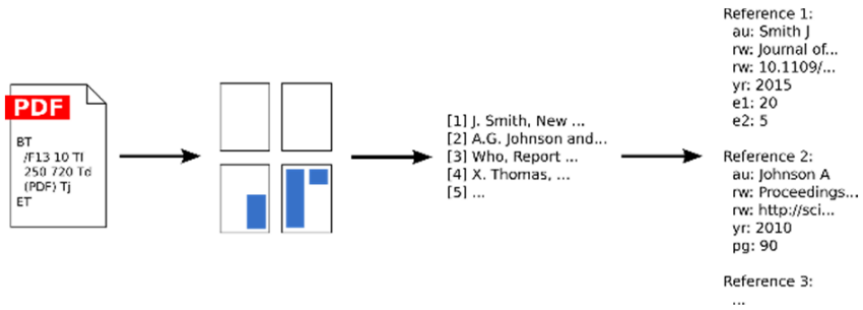

Figure 2.1: A common workflow for parsing citations from a document. 1. The bibliography section is identified. 2. The bibliography is split into individual citation strings. 3. The metadata is retrieved from each individual citation string by a citation parser. [1]

2.2.2 Sequence Labelling

Supervised machine learning is one established approach to tackling citation parsing. In supervised machine learning citation parsing can be formally defined as a sequence labelling problem [1]. In a sequence labelling problem the input is a sequence of objects or features. The aim is to assign a label to each input object taking into account both the object itself and the dependencies between neighbouring objects [1].

In citation parsing, the sequence of objects are individual words, punctuation or blank spaces contained within the citation string. The individual objects are commonly called tokens. The first task of a citation parser is to split a given citation string C into a sequence of individual tokens $\{t_1, t_2, .. t_n\}$. This process is called tokenization. The following citation string:

D. Foo, Abbreviations from Bar to Baz, Phys. Rev. 9, 34(2019)

split into individual tokens would be:

C = {D, ., , Foo, „ , Abbreviations, , from, , Bar, , to, , Baz, „ , Phys, ., , Rev, ., , 9, „ , 34, (, 2019,)}

Once a string has been tokenized the second task of the citation parser is to assign to each individual token the correct label from a set of classes $C = \{c_1, c_2, .. c_n\}$. Commonly these labels correspond to the desired field types, with "other" being used for any token that is not part of a field.

Table 2.2 shows the same citation with tokens and their corresponding label. Blank spaces are removed for brevity. AU-FN refers to the author's first name and maps to the author field. OTH refers to the "other" field and is used for blank spaces and punctuation not belonging to a particular field.

Unfortunately, different citation parsers can assign different labels to individual tokens. Biblio [45], Parscit [25] and Science Parse [46], three popular citation parsing tools, label the author's full name as one field while other tools label first name, middle name and surname with distinct labels. Furthermore, different training datasets can also assign different labels to different fields with some being more or less fine-grained than others.

After predicting the labels for each token, tokens with a common field can be concatenated. Figure 2.2 summarises the overall flow of a sequence labeller.

Token	Label
D	AU-FN
.	AU-FN
Foo	AU-SN
,	OTH
Abbreviations	TITLE
from	TITLE
Bar	TITLE
to	TITLE
Baz	TITLE
,	OTH
Phys	JOURNAL
.	JOURNAL
Rev	JOURNAL
.	JOURNAL
9	VOLUME
,	OTH
34	ISSUE
(	OTH
2019	DATE
)	OTH

Table 2.2: A citation string split into it's respective tokens and assigned label.

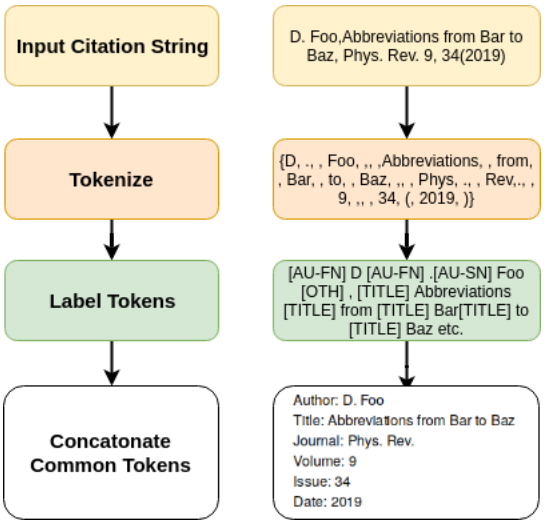

Figure 2.2: The workflow of a sequence labeller. 1. Citation String is tokenized. 2. Labels are predicted for individual tokens. 3. Common token labels are concatonated.

2.2.3 Training Data

In order to train a supervised machine learning citation parser labelled tokens such as those shown in Table 2.2 are needed. In practice, this training data is usually stored as labelled XML and then converted into a sequence of labelled tokens. Figure 2.3 shows the corresponding XML for the labelled tokens in Table 2.2. Again, the exact nature of the XML tags may vary depending on the citation parser being used.

```
<bibl>
  <author>D. Foo</author>
  <title>Abbreviations from Bar to Baz</title>
  <journal>Phys. Rev.</journal>
  <volume>9</volume>
  <issue>34</issue>
  (<date>2019</date>)
</bibl>
```

Figure 2.3: An example of labelled XML which is used as input to a train a ML citation parser.

Another common format for training data follows the CoNLL convention. In this format, each line of the file contains a token followed by the corresponding label. A blank line is used to separate individual citations. The citation:

D. Foo, Abbreviations from Bar to Baz, Phys. Rev. 9, 34(2019)

in CoNLL format is shown in Table 2.3.

2.3 Evaluation Methods

The strength of a citation parser is typically evaluated using the metrics recall, precision and F1. These are defined as follows:

- Recall = $\frac{TruePositive}{TruePositive+FalseNegative}$ [47]
- Precision = $\frac{TruePositive}{TruePositive+FalsePositive}$ [47]
- F1 = $\frac{2*Precision*Recall}{Precision+Recall}$ [47]

The terms positive and negative here refer to a citation parser's predicted label for a given token or field and the terms true and false refer to whether that token or field has been labelled correctly.

Recall can be viewed as the proportion of actual positive labels that were correctly predicted. Precision can be viewed as the proportion of predicted positive labels that were correct. F1 combines recall and precision into a single metric. It is the most commonly reported figure for measuring the performance of citation parsing tools.

D author
. author
Foo author
Abbreviations title
from title
Bar title
to title
Baz title
Phys journal
. journal
Rev journal
. journal
9 volume
34 issue
2019 date

Table 2.3: An example of a citation string in CoNLL format used to train a supervised ML citation parser.

The evaluation of a citation parser is commonly subdivided by field as shown in Table 2.4. Sometimes both macro and micro averages are reported.

Field	Precision	Recall	F_1
Author	0.94	0.94	0.94
Journal	0.16	0.70	0.26
Volume	0.76	0.19	0.31
Pages	0.91	0.58	0.71
Date	0.84	0.64	0.72
Macro-Average	0.72	0.61	0.59
Micro-Average	0.79	0.74	0.76

Table 2.4: Sample evaulation metrics subdivided by field.

To calculate the macro-average, the metric for each field is calculated independently and the mean of all fields is calculated. In Table 2.4, there are five fields. The sum of all entries for F1 is 2.94 and the macro-average is $2.94/5 = 0.59$. The macro-average may not always be the most useful metric due to class imbalance [48]. In citation strings the classes are highly likely to be imbalanced with fields such as *author* and *title* occurring more often than *URL* or *Note*.

Micro-average is calculated by aggregating the contributions of all fields and dividing by the total number of contributions. For example, in calculating precision, we would sum the number of True Positives for every field

and divide by the sum of the total number of True Positives and False Positives. Typically, in citation parsing this figure would be higher than the macro-average, again, due to class imbalance.

The F1 micro-average would be a commonly reported single figure used to represent the overall strength of a given citation parser. Both token-level and field-level results for recall, precision and F1 are reported. Preference is usually given to reporting the more meaningful field-level result however, some papers report both.

Chapter 3

Related Work

CiteSeer, first developed in 1998, was an early attempt to create an automatic citation indexing system using citation parsing [49]. In subsequent years citation string parsing has become an established research problem and a number of different approaches have been proposed. These include: regular expressions, knowledge bases, template matching, and supervised machine learning.

3.1 Regular Expressions, Knowledge Bases and Template Matching

Early efforts to the problem of citation parsing, such as that by Kunnas [50], used regular expressions. As Tkaczyk et al. [1] and Zhang et al. [19] comment, regular expressions can work well when the data has little noise, the number of citation styles is small and the citation styles are predefined. However in real-world data these conditions are rarely met and further, regular expressions lack adaptability, are expensive to maintain and don't scale well [19].

In an effort to address these shortcomings a number of authors combined regular expressions with knowledge bases [51, 52, 53]. A knowledge based approach involves populating the system with knowledge from relevant and available data sources. This knowledge may include journal titles and authors' names. During the citation parsing process a field extracted from the citation string can then be matched against the known knowledge base.

Constantin et al. [53] created PDFX, a rule-based system which deconstructs a given PDF into an XML document describing the document's logical structure. Evaluating across three different datasets they report an average F1 score of 0.74 for reference parsing. Their results did not include documents created using OCR.

Cortez et al. [52] used a knowledge based approach to create FLUX-CiM. They conducted experiments matching seven fields: author, title, date, jour-

nal, volume, issue and page, across three domains: health science, computer science and social science, and they reported that FLUX-CiM achieves precision and recall above 94%.

Heckmann et al. [51] detail a knowledge based approach which use Markov logic networks (MLN) [54]. They evaluate their model on a novel data set featuring sparse and noisy data and they report a 24.8% increase in F1 score (0.88) over the popular CRF ML approach.

Summarising these knowledge based approaches Tkaczyk et al. [1] comment that they work best when the knowledge base forms a closed set. However, once again, they can be difficult to maintain.

A third approach to the problem of citation parsing is template matching. This approach involves matching a given citation to a database of known citation templates. This step is usually followed by applying regular expressions or template-based rules.

A template-based approach performs well when the number of citation styles is limited. Multiple authors report field-level accuracy of above 90% when experimenting with 22 or fewer citation styles [35, 55]. Hsieh et al. [56] also report a decrease in the average field error rate by 70% (2.24% vs 7.54%) when compared with the popular machine learning CRF approach.

However, as Chen et al. comment "the query processing performs better when the template database is consistent with the test data [35]." With BibPro, a citation parser developed using a sequence alignment tool called BLAST (Basic Local Alignment Search Tool), they reported the highest average field-level accuracy when evaluating on a dataset containing only six distinct citation styles [35].

With thousands of citation styles in use the challenge for a template-based approach is scalability [25, 13]. ParaTools [57] maintains 400 templates to match citation strings but adding new templates and maintaining existing templates remains cumbersome.

Another significant challenge associated with a template-based approach is that errors within the citation string, caused by human error or OCR, may produce citation strings that do not strictly adhere to the template format. Council et al. comment that "the lack of portability makes the approach unsuitable for high volume data processing [25]."

3.2 Machine Learning

3.2.1 An Overview

Figure 3.1 shows that the number of papers which have adopted a ML approach to citation parsing greatly outnumbers those who have used non-ML methods. Since 2010, 77% (24) of 31 reviewed published papers surveyed in the area of citation parsing have adopted a ML-based approach. This

perhaps reflects the growing consensus around the strengths of using ML methods.

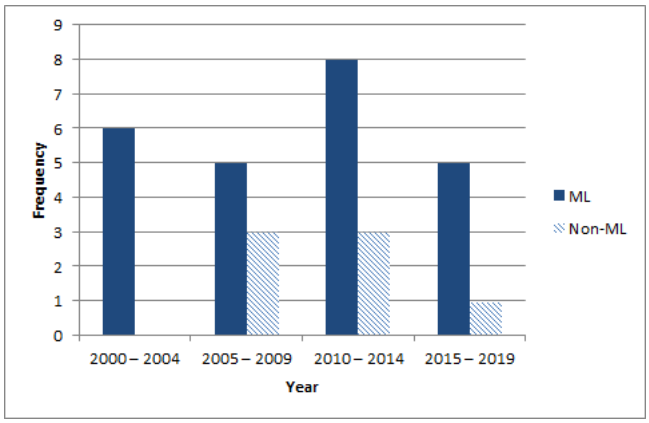

Figure 3.1: The number of papers which have adopted a ML-based and a non ML-based approach to citation parsing between 2000 and 2019

The four most common ML approaches are Support Vector Machines (SVM) [19, 58] Hidden Markov Model (HMM) [21, 22, 23, 24], Conditional Random Fields (CRF) [31, 18, 25, 26, 27, 28, 29, 30] and deep-learning [13, 32].

Figure 3.2 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.

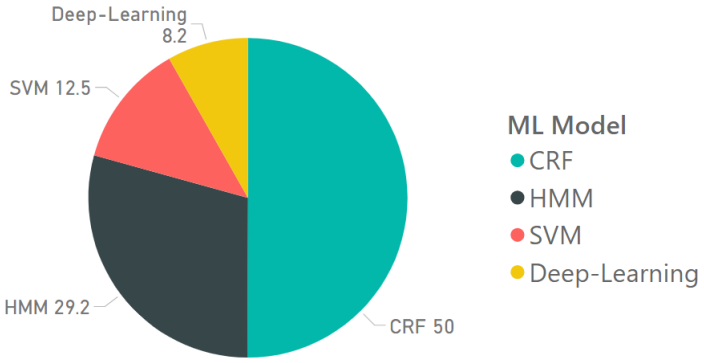

Figure 3.2: The proportion of ML papers which used SVM, HMM, CRF and Deep-Learning

Figure 3.3 shows how the popularity of certain 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.

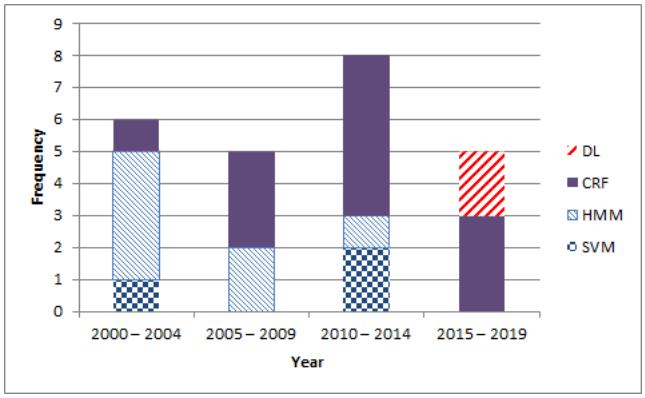

Figure 3.3: The changing popularity of ML citation parsers

Unlike template matching or regular expressions a ML approach does not require expert knowledge to keep the citation parser up to date. Tkaczyk et al. [1] showed that the performance of ML models can be improved by retraining a ML model on task-specific data. Reviewing ML-based citation parsers they showed that the top three performing tools improved their F1 score by an average of 10% when re-trained on task-specific data.

The implication is that in order to improve a models performance on a particular citation style, or domain-specific language, a ML model only needs to be re-trained on more up-to-date data. This removes the constraint of having to create a new template or adapt an existing knowledge base. As Tkaczyk comments "it is comparatively easy to make sure the models are up to date by repeatedly retraining them on newer data [1]."

3.2.2 Support Vector Machines and Hidden Markov Models

SVM is a general-purpose ML classifier that has been applied to the task of classifying tokens within a citation string. Okada et al. [58] combined SVM and HMM. Using 5-fold cross validation with 4,651 citations, they found that their combined approach had recall 14% higher than an SVM approach (0.988 vs 0.974).

Zhang et al. [19] compared structural SVM with conventional SVM. Structural SVM utilizes the contextual information contained within neighbouring features. Using 600 references for training and 1800 references for testing taken from PubMed [59], they found that structural SVM outperformed SVM on field-level accuracy (96.95% vs 95.59%). Although the results of Zhang and Okada are promising, Zhang reports that SVM performed worse than the state-of-the-art CRF model [19].

Another ML approach to citation parsing is Hidden Markov model (HMM). HMM is a probabilistic model which assumes that the system to be modelled is a Markov process with hidden states.

Hetzner [24] presents a simple HMM-based model which utilizes the widely available Viterbi algorithm. Trained and evaluated on the popular Cora dataset the model achieves comparable performance to other benchmark HMM models however their results are inferior to the CRF-based work of Peng and McCallum [60].

Yin et al. [61] describes a variation of HMM model which considers words' bigram sequential relation and position information. Using 4-fold cross validation on 736 labelled citations, they note a 3.6% improvement in the performance of Bigram HMM compared with regular HMM (F1 0.991 vs F1 0.868).

Ojokoh et al. [62] made further improvements utilizing a trigram HMM. Using 4-fold cross validation with three different training datasets ranging in size from 300 to 712, they reported recall, precision and f1 scores of above 95%. Interestingly they report that increasing the size of the training dataset from 275 citations to 537 citations only improved accuracy by 0.07%.

3.2.3 Conditional Random Fields

Among the different ML approaches CRF is by far the most popular method with 50% of 24 surveyed ML papers adopting a CRF-based model.

In 2018, Tkaczyk et al. [1] carried out a comparison survey of ten existing citation parsing tools. They reported that the highest three performing tools, GROBID (F1 0.89) [26], CERMINE (F1 0.83) [18] and Parscit (F1 0.75) [25], all used a CRF algorithm. Unfortunately, this comparison survey, did not include the two existing deep-learning approaches. This was due to missing resources, in the case of Neural Parscit [13] and because the work was carried out after the survey was completed, in the case of Rodrigues et al. [32].

GROBID [26] is a CRF based tool that can parse individual citation strings as well providing wider functionality in document processing. With it's performance in the recent comparison survey of Tkaczyk et al. [1] it can be considered the state-of-the-art in CRF based citation parsing. GROBID is trained on 7,800 labelled citations and evaluating on the CORA dataset Lopez reported a field-level accuracy of 95.7%.

CERMINE [18] is another CRF citation parser that exists as part of a larger document processing tool. It is able to extract citation strings, their metadata and the structured content of a document's body directly from a PDF. CERMINE won best performing tool award in the 2015 Semantic Publishing Challenge [63]. This included tasks related to parsing citations.

Inspired by Peng and McCallum [60], Parscit [25] is an open-source CRF based tool that can locate, parse and retrieve the context of citation strings. Parscit is deployed as part of CiteSeerX [64], a large computer science digital library. Using 10-fold cross validation on the CORA dataset Parscit showed 4.4% improvement in comparison to Peng and McCallum's earlier work (micro-average F1 0.95 vs macro-average F1 0.91).

3.2.4 Deep Learning

Advances have been made in recent years in the application of deep learning techniques to a variety of Natural Language Processing (NLP) tasks including sequence-labelling. The state-of-the-art architectures for sequence labelling include a CRF prediction layer [65], word-embeddings and character-level word-embeddings. They are trained either with Convolutional Neural Networks (CNNs) [66] or Recursive Neural Networks (RNNs) using Bidirectional Long-Short Term Memory (BiLSTM) layers [67].

Rodrigues et al. [32] apply and compare the architectures of Lample et al. [67] and Ma and Hovy [66] to the task of reference mining. They define reference mining as the "detection, extraction and classification of bibliographic references [32]." Their model was trained and evaluated on citations extracted from a corpus of literature on the history of Venice. Word embeddings were pre-trained using Word2Vec [68] on the entire publications from which they extracted citations and the model was trained on 40,000 citations. Extensive tuning was undertaken. Their final model outperformed a CRF baseline by 7.03% achieving an F1 of 0.896.

Prasad et al. [13] also examined how well a deep-learning approach would handle the task of citation parsing. They carried out extensive model experimentation and tuning using both word-embeddings and character-based word-embeddings. Their final model deployed a Long Short-Term Memory (LSTM) neural network with a layered CRF over the output. In comparison against Parscit [25], a CRF-only based citation parser, they reported a significant ($p < 0.01$) improvement.

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" [32], their results are promising and highlight the potential of a deep-learning approach.

3.2.5 Meta-Learning

Aware that different citation parsing tools can perform better or worse depending on the citation string given and the fields to be extracted, Tkaczyk et al. explored a meta-learning approach to citation parsing [33]. They presented ParsRec, a recommender-system which suggests the best citation parser for a given citation string.

They explored two approaches to meta-learning recommendations. The first learnt the best citation parsing tool for a given citation string whilst the second, learnt the best tool for a given field. Evaluating on 105,000 references from the Chemistry domain, they found that the second approach achieved a 2.6% increase in F1 (0.909 vs. 0.886, $p < 0.001$) over GROBID, the best individual parsing tool.

3.3 Comparing Approaches

Drawing a fair comparison between existing citation parsing tools is challenging for a number of reasons. These reasons include the chosen evaluation metrics, the different capabilities for fine-grained extraction and the evaluation datasets in use.

3.3.1 Evaluation Metrics

There is no single established metric for evaluating the performance of a citation parser. In a survey I conducted of 31 papers on the topic of citation parsing, between 2000-2018, evaluation metrics reported included:

- Precision
- Recall
- F1
- Field-level Accuracy
- Token-level Accuracy
- Word-level Accuracy
- Micro Average
- Macro Average
- Field-error Average

The most commonly reported metric was F1 at 55%. F1 was reported alongside recall and precision in 45% of papers. Average values of F1 were reported as micro-average, macro-average or just "average" with not every author being explicit in which average they were using.

The second most popular metric reported was accuracy at 51%. Here, some authors made the explicit distinction between token-level and field-level accuracy. Other authors reported a single figure without making their choice of average explicit. Note that the totals for F1 (55%) and accuracy (51%) don't add up to 100% as some authors reported both metrics. The lack of a single chosen evaluation metric makes comparing the results from different approaches challenging.

3.3.2 Fine-Grained Extraction

The exact number of fields that each citation parser is either tested on or, is capable of extracting, varies. Table 3.1 summarises the fields extracted by six popular open-source citation parsing tools. The most fine-grained tools,

Tool	Approach	Extracted Fields
Biblio	Regular Expressions	author, date, editor, genre, issue, pages, publisher, title, volume, year
BibPro	Template Matching	author, title, venue, volume, issue, page, date, journal, booktitle, techReport
CERMINE	CRF	author, issue, pages, title, volume, year, DOI, ISSN
GROBID	CRF	authors, booktitle, date, editor, issue, journal, location, note, pages, publisher, title, volume, web
Parscit	CRF	author, booktitle, date, editor, institution, journal, location, note, pages, publisher, tech, title, volume
Neural Parscit	Deep-Learning	author, booktitle, date, editor, institution, journal, location, note, pages, publisher, tech, title, volume

Table 3.1: The approach and extracted fields of six popular open-source citation parsing tools.

Parscit [25] and Neural-Parscit [13], can extract twelve different fields whilst the least fine-grained tool, CERMINE [18], can extract eight different fields.

The varying number of fields a citation parser can extract adds to the complexity of comparing tools. For instance, GROBID [26] makes the distinction between *title* and *book-title* whilst CERMINE [18] simply reports *title*. It is possible that the results of GROBID in reporting *title* will be below that of CERMINE as GROBID has the added task of distinguishing between a book-title and an article title. However, perhaps the lower performance on *title* may be acceptable in view of the finer-grained output.

3.3.3 Evaluation Datasets

Comparing different citation parsing tools is also made difficult by the fact that different authors use different datasets for evaluation and these datasets vary greatly in size, domain and homogeneity. Table 3.2 summarises the evaluation datasets used by nine citation parsers. Of these nine examples, 44% report results from a single evaluation dataset, 44% of authors report

results for multiple datasets and 22% of authors use an amalgamation of two or more datasets.

Citation Parser	Evaluation Dataset	Size	Domain
Structural SVM [19]	PubMed	1800	Health Science
HMM [24]	Cora	142	Computer Science
Bigram HMM [61]	ManCreat	712	NA
BILBO [69]	Umich	NA	Chemistry, Biology, Humanities, Technology
Trigram HMM [62]	Cora	500	Computer Science
	Flux-CiM	300	Computer Science
	ManCreat	712	NA
Parscit [25]	Cora	200	Computer Science
	Flux-CiM	300	Computer Science
	CiteseerX	200	Artificial Intelligence
GROBID [26]	PubMed	90079	Health Science
	Cora	1295	Computer Science
CERMINE [18]	Cora + Citeseer	3438	Computer Science
Neural Parscit [13]	Cora	500	Computer Science
	Cora + FluxCiM + ICONIP + humanities	1053	Cross-Domain

Table 3.2: Evaluation datasets used by nine citation parsing tools broken down by size and domain.

As well as the fact that there is no consensus on a gold standard evaluation dataset a number of other issues surround the datasets which do exist. Table 3.3 summarises popular evaluation datasets along with their size and domain.

A primary problem with the datasets in Table 3.3 is their homogeneity – every citation in the dataset coming from a single domain or sub-domain. The aim of a citation parser is to perform well on citations from a variety of domains and across a wide range of citation styles and types. Therefore, evaluating their performance on a dataset from a single domain will not fairly reflect how a citation parser will perform across the full range of citations that exist today. A number of authors [13, 18, 25, 69] have tried to address this issue by evaluating on multiple datasets across two, or sometimes, three domains.

The Cora dataset in Table 3.3 is one of the most widely used datasets in the field of citation parsing but unfortunately, as a number of authors have commented, it contains significant weaknesses [37, 13, 25]. Firstly it is

Dataset Name	Size	Domain
Cora [70]	1295	Computer Science
Citeseer [49]	1563	Artifical Intelligence
Umass Citation Field Extraction Dataset [37]	1829	Physics, Mathematics, Computer Science, Quantitative Biology
FLUX-CiM CS [71]	300	Computer Science
FLUX-CiM HS [71]	2000	Health Science
PubMed [59]	29000000	Biomedical
GROTOAP2 [72]	6858	Biomedical, Computer Science
CS-SW [71]	578	Semantic Web Conferences
Venice [32]	40000	Humanities
ManCreat [62]	712	NA

Table 3.3: Summary of existing labelled citation datasets.

homogeneous, with citations coming exclusively from the Computer Science domain. It is also "small" and only has labels for "coarse-grained fields" [37]. For example, the author field does not label each author separately. Prasad et al. echo these comments saying that a "shortcoming of the field is that the evaluations have been largely limited to the Cora dataset, which is... unrepresentative of the multilingual, multidisciplinary scholastic reality" [13].

A final problem is that training and evaluating datasets often are taken from the same domain. Table 3.4 lists six examples of citation parsers which were trained and evaluated on the same domain or, in some instances, dataset. The problem with this approach is that a single domain will not fairly reflect the diversity of citation styles in existence. Furthermore, performing well on one homogeneous dataset or domain will not necessarily reflect a tool's performance on unseen citations from a different domain.

As an experiment to investigate this point I retrained GROBID [26] using 70% of it's original training data. This model was then evaluated on the remaining 30% of it's unused training data as well as the Cora dataset. The retrained Grobid had a micro-average F1 of 0.951 when evaluated on the remaining 30% of it's training data and a micro-average F1 of 0.745 when evaluated on Cora. This is a performance drop of 21.6% and highlights how there can be large fluctuations in a model's performance depending on how it is evaluated.

Citation Parser	Training Domain	Evaluation Domain
Structural SVM [19]	Health Science	Health Science
HMM [24]	Computer Science	Computer Science
Deep Mining [32]	Humanities	Humanities
Bigram HMM [61]	Cross Validation	Cross Validation
Trigram HMM [62]	Cross Validation	Cross Validation
CERMINE [18]	Computer Science & Health Science	Computer Science

Table 3.4: Examples of ML citation parsing tools which are trained and evaluated using datasets from the same domain.

3.3.4 Comparing Approaches Conclusion

Drawing comparisons between tools which have been evaluated on different datasets is debatable, at best. Fortunately in 2018, Tkaczyk et al. [1] carried out a comparison survey of ten existing citation parsing tools, six machine-learning based tools and four non machine-learning based. Evaluated on 64,495 references taken from the Chemistry domain, the results show that on average ML-based tools outperform non ML-based approaches by 133% (F1 0.77 for ML-based tools vs F1 0.33 for non-ML based tools). Furthermore the lowest performing ML-based citation parser, Anystyle-Parser [73], outperformed the best performing non-ML based citation parser, Biblio [45], by 28.6% (F1 0.54 vs F1 0.42).

Tkacyzk et al. [1] also reported that the highest three performing tools, GROBID (F1 0.89) [26], CERMINE (F1 0.83) [18] and Parscit (F1 0.75) [25], all used a CRF algorithm. Unfortunately, they were not able to include the two deep-learning methods in their survey. This was due to missing resources, in the case of Neural Parscit [13] and because the work was carried out after the survey was completed, in the case of Rodrigues et al. [32].

3.4 Training Datasets

Table 3.5 summarises the size and domain of the training datasets used by eight ML citation parsing tools.

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 [59] 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

Citation Parser	Training Dataset	Size	Domain
GROBID [26]	NA	7800	NA
CERMINE [18]	GROTOAP2	6858	Computer Science & Health Science
Structural SVM [19]	PubMed	600	Health Science
HMM [24]	Cora	350	Cora
Bigram HMM [61]	ManCreat	712	NA
Trigram HMM [62]	Cora + FluxCiM + ManCreat	1512	Computer Science
Deep Mining [32]	Venice	40000	Humanities
SVM + HMM [58]	IEICE Transactions on Fundamentals of Electronics, Communications and CS	4651	Computer Science

Table 3.5: Training datasets of eight ML citation parsing tools.

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. [32] who have used a deep-learning approach and a training dataset of 40,000 citations, the remainder of 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 potential benefits of having more training data available [37, 38] and the limitations of existing datasets [37, 32, 13].

In an effort to address these problems, in 2018 Ryan [74] created a dataset with 400,000 unique references in fifty different citation styles using web scraping. However, attempts to further develop the size and diversity of this dataset were hampered by computational and resource constraints. Unfortunately no record was given that any existing ML citation parser was successfully retrained using this dataset.

Deep-learning has led to huge advances in recent years in a variety of fields including image classification [75], sentiment analysis [76], image recognition [77], video translation [78] and text generation [79].

For image-classification, prior to deep-learning, small, hand-crafted datasets were typically used to train models. Datasets such as NORB [80], Caltech-101/256 [81] and CIFAR-10/100 [82] were of the order of tens of thousands. To produce CIFAR-10, students were paid to classify ten different objects, with the final dataset containing 6000 images of each object [82].

The advent of new, larger datasets such as LabelMe [83], which contains

hundreds of thousands of images, and ImageNet [84], which contains over 15 million labelled images, enabled the application of deep-learning in this field. In the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) Krizhevsky et al. [75] achieved a winning error rate of 15.3% using a deep-learning approach for classifying images. This was a 41.6% improvement on the second best entry which used non deep-learning methods.

The size of the training datasets available and used for citation parsing are in stark contrast to those available in other areas of machine learning. It remains to be seen the effect a large and diverse training dataset can have on this open research problem.

Chapter 4

Methodology

4.1 Citation Style Languages

In order to create the 1 Billion Citation Dataset a particular XML-based language called Citation Style Languages (CSL) will be used. A brief introduction to CSL is given below. This is then followed by a detailed description on how CSL was used to create the 1 Billion Citation Dataset.

4.1.1 Introduction

Reference management tools such as Zotero [44], Mendeley [85] and EndNote [86] are used to help individuals to organize their research. They are also used to automatically generate citations and bibliographies in any required citation style. However, in order to format references in the desired style these programs must contain some description of each citation style in a machine-readable format. Citation Style Language (CSL), an XML-based language, can be used to provide exactly this, namely, a machine-readable description of citation styles and bibliographic format [2].

Figure 4.1 provides an overview of the CSL ecosystem. There are four components that are used by a CSL processor to generate a citation. These are: an item's style, a locale file, metatdata information and citing details.

4.1.2 Citation Styles

In CSL there are two types of citation styles: independent and dependent. An independent CSL style defines the format of a citation. It provides details about the citation's layout, punctuation and overall structure [87]. It will answer questions such as: should an author's first name be provided or just their initial, should a DOI be included, where does the date go in the citation etc. It also provides style metadata, information which describes the style itself. This could include the name of the person, who created the style, the date the style was created etc.

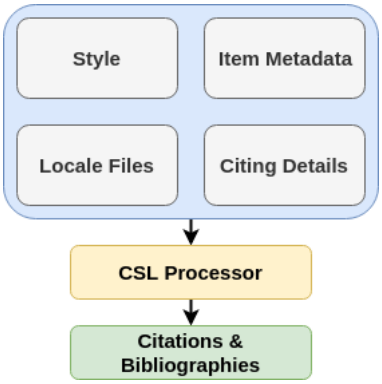

Figure 4.1: An overview of the CSL Ecosystem. An item's style, locale file, metadata and citing details are all combined in a CSL processor to produce a citation string or bibliography. [2]

Dependent styles on the other hand contain only style metadata. They are used as pointers to independent styles and are useful when multiple CSL styles share the same citation format [87]. If multiple journals use the same citation format, each individual journal can have their own dependent style, which then points to a single independent style which defines the citation layout. This has the benefit that if a publishing group decide to change their citation format for a group of journals, only a single, independent style, will have to be updated.

4.1.3 Locale Files

Locale files are used to define language-specific phrases in a citation string. For example, the following citation string makes use of a US English locale file:

Hartman, P., Bezos, J. P., Kaphan, S., & Spiegel, J. (1999, September 28). Method and system for placing a purchase order via a communications network. Retrieved from https://www.google.com/patents/US5960411

Changing the locale file to German will alter the text "Retrieved from" to the German "Abgerufen von". The complete citation string will then appear as:

Hartman, P., Bezos, J. P., Kaphan, S., & Spiegel, J. (28. September 1999). Method and system for placing a purchase order via a communications network. Abgerufen von https://www.google.com/patents/US5960411

Each language has it's own locale file and these enable CSL styles to be largely language independent [87].

4.1.4 Item Metadata

Item metadata stores the bibliographic details of the entry you wish to cite. This may include the author's name, the journal title, the date, the page number etc. Reference managers typically have their own method of storing an item's metadata however common formats include JSON, BibTeX and Research Information Systems (RIS).

4.1.5 Citing Details

Citing details refer to context-specific information. For example, if an item has already been cited in a document this may mean that future citations of the same work are given in a more compact form. Citing details are not relevant for creation of the 1 Billion Dataset but are included here for completeness.

4.1.6 CSL Processor

Each of the four items – citation style, locale file, item metadata and citing details – are fed into a piece of software called a CSL processor. The CSL processor then generates the desired citation string and/or bibliography. Many reference management tools make use of open-source CSL processors such as citeproc-js [88].

4.1.7 CSL Example

Figure 4.3 shows an example of a small independent CSL style. In reality CSL styles are much longer. In Figure 4.3 the style's metadata is contained within the <info> tags. For example, the name of the style is given as *Example Style* and the style's creator is listed as *John Doe*. Next, the locale file is listed as English: <locale xml:lang="en">. The remainder of the CSL can be viewed in three parts:

- <macro>
- <citation>
- <bibliography>

Macros are reusable pieces of XML. The macro in Figure 4.2 details that an author's name should appear as an initial followed by a full stop (M. Grennan). The remainder of the macros are removed for brevity.

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

Figure 4.2: An example of a macro contained within a CSL independent style.

<Citation> details how an in-text citation should appear. Here the code generates citations like: *M. Grennan et al., 2019*. Finally, the XML within <bibliography> details how a reference in a bibliography should appear.

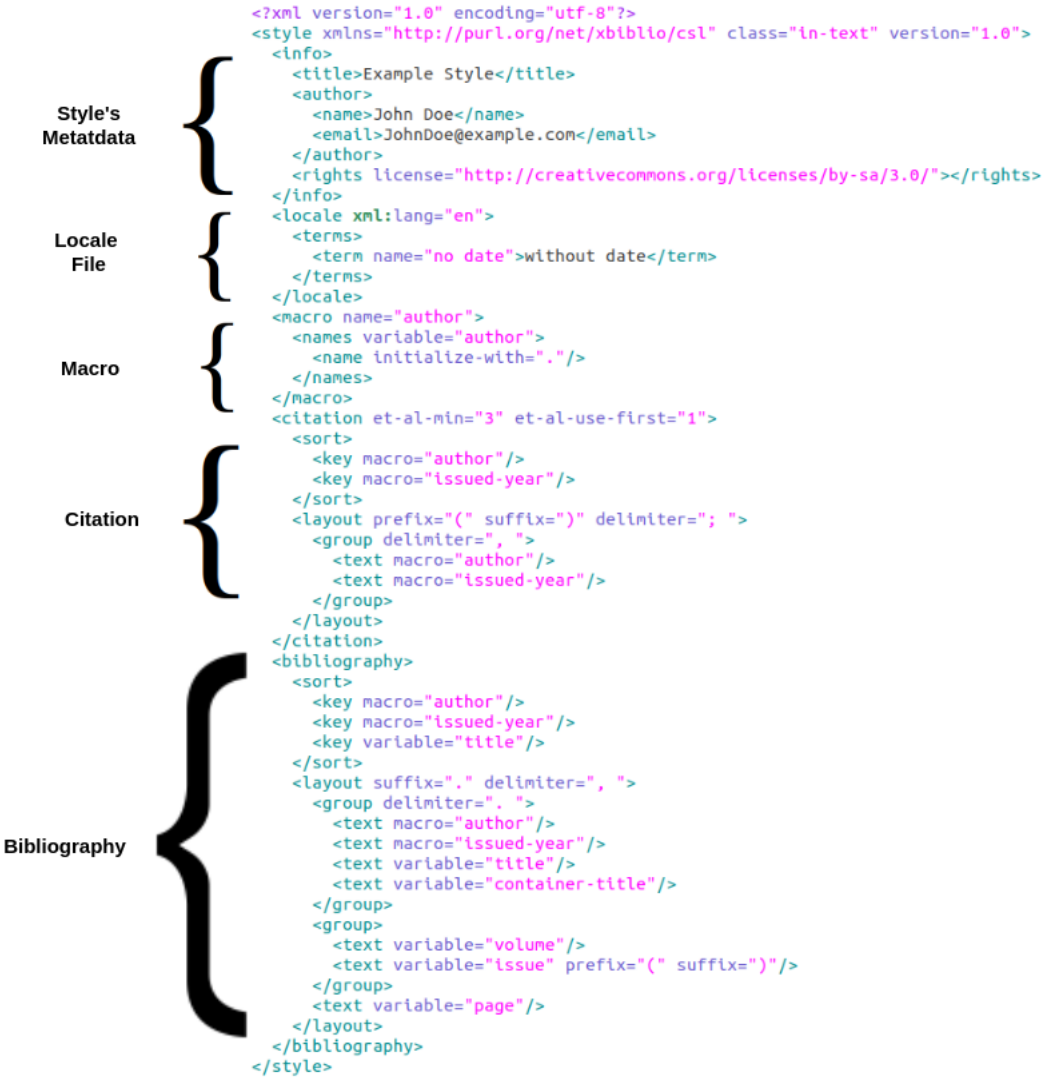

Figure 4.3: An example of an independent CSL style.

4.2 Creation of 1 Billion Citation Dataset

4.2.1 Introduction

As shown in Figure 4.4, combing an XML citation style, an item's metadata and locale file in a CSL processor will produce a citation string. In Figure 4.4 the following citation string is produced: *M. Grennan, 1st August 2019, The 1 Billion Dataset.*

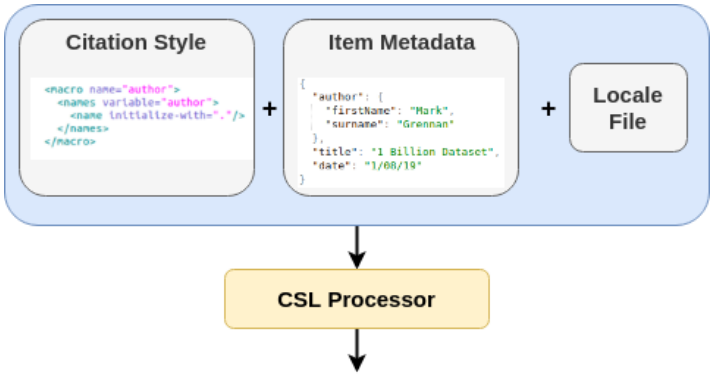

Figure 4.4: Combining a CSL Style, an item's metadata and locale file in a CSL processor will produce a citation string.

However, training data for ML citation parsers must be labelled XML citation strings such as that shown in Figure 4.5. In order to create this training data the XML citation styles were edited. These edited citation styles, along with a locale file and an item's 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>
```

Figure 4.5: An example of labelled XML which is used as input to a train a ML citation parser.

Figure 4.6 gives a high-level overview of the process. What follows is a description of the steps undertaken to create the 1 Billion Dataset along with the motivation behind each step.

4.2.2 Editing Citation Styles

The aim in creating the 1 Billion Citation Dataset was to create a dataset that contains as many different citation styles as possible. Although some styles are more commonly used than others it was decided to include equal

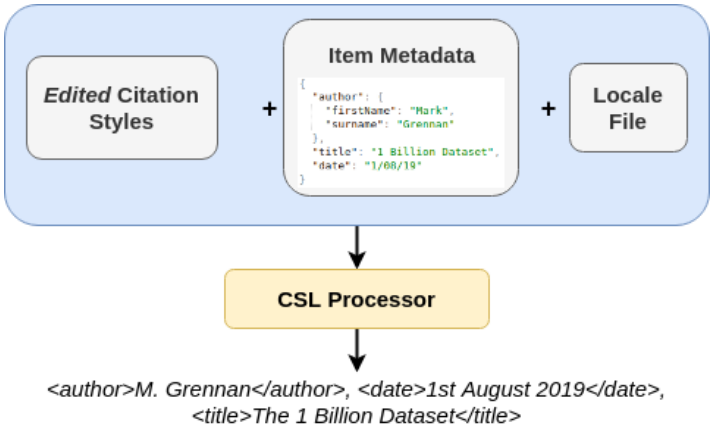

Figure 4.6: An overview of the process of creating the 1 Billion Citation Dataset. The citation styles were edited and combined with citation metadata records and a locale file in a CSL processor. The final output is labelled XML citation strings.

numbers of each citation style. In this way, a researcher should be able to use the 1 Billion Dataset to create their own training data specific to their needs. Should they require training data with a particular citation style, or set of styles, the aim is that these styles will all be contained within the 1 Billion Dataset. With this in mind 1,564 independent XML citation styles were obtained from the official CSL repository on github [89].

As described in the Introduction 4.2.1, combing each XML citation style with an item's metadata in a CSL processor would generate a citation string. However, we require a dataset of *labelled* citation strings. The first task in creating the 1 Billion Citation Dataset was to edit the 1,564 XML citation styles so that each field (author, title etc.) would contain a prefix tag (<author>, <title> etc.) and a suffix tag (</author>, </title> etc.).

Table 4.1 gives examples of fields both before and after the prefix and suffix tags were added. In each citation style the field name was either contained within a label called *name* or *variable*. As an example in Table 4.1, the publisher field is contained within a text tag and is labelled with *variable*. The date field is contained within a date-part tag and is labelled with *name*.

The author field is slightly more complicated. Different citation parsing tools require slightly different formatting for their training data. Some tools such as GROBID [26] require that all the authors names are contained within a single author tag. For example, here the authors *M. Grennan* and *U. McMenamin* are contained within an outer *author* tag:

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

Other tools such as Biblio [45] and Parscit [25], require individual au-

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>"/>
title	<text variable="title">	<text variable="title" prefix="<title>" suffix="</title>"/>
issue	<text variable="issue" prefix="(" suffix=")"/>	<text variable="issue" prefix="(<issue>" suffix="</issue>)"/>

Table 4.1: Original CSL tags and CSL tags after a prefix and suffix tag has been added.

thors to be encapsulated within their own tag. Here, *M. Grennan* and *U. McMenamin* are contained within their own individual *name* tags. Both are encapsulated within an outer *author* tag:

<author><name>M. Grennan</name>, <name>U. McMenamin</name></author>

Finally, some tools tag an author's first name, middle name and/or surname separately. For example, here *M.* is contained within a *firstname* tag and *Grennan* is contained within a *surname* tag:

<author><firstname>M.</firstname> <surname>Grennan</surname></author>

In an effort to make the 1 Billion Dataset as widely usable as possible an author's first name, middle name and surname were given separate tags. This means that, with only a slight modification, the 1 Billion Dataset can be converted into all of the previously mentioned formats. For instance, should a citation parser require a single author tag the inner tags for an author's first name, middle name and surname can just be removed.

In the 1 Billion Dataset, 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. Figure 4.7 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

```
<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>
```

Figure 4.7: 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.

Before editing, the macro in Figure 4.7 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>

Should a citation contain multiple authors 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>

A similar approach is taken with macros for editor.

4.2.3 Locale File

CSL locale files, in combination with the XML citation styles, enable citations to be created in many different languages. In creating the 1 Billion Dataset the US-English locale file was used.

4.2.4 Item Metadata and Crossref

The requirements for the 1 Billion Dataset are that it is diverse and large. The diversity of citation styles has been addressed. 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 [90]. CrossRef is a not-for-profit organisation that collates, tags and shares metadata on scholarly publications. Their records contain over a hundred billion items from a diverse range of academic fields.

677,000 records were obtained from Crossref using their public API. Crossref's *random_doi* method was used to obtain random records. These records were returned in JSON format in a single 2.2Gb file.

Each of these records was then converted into a labelled citation string in each of the 1,564 different citation styles. So for every item obtained from Crossref it appears 1,564 times in the 1 Billion Dataset.

Finally, when implementing this process the 2.2Gb file from Crossref was split into 219 smaller files each of size 10Mb. Each 10Mb input file produced a 2Gb output csv file. An overview of this process is shown in Figure 4.8.

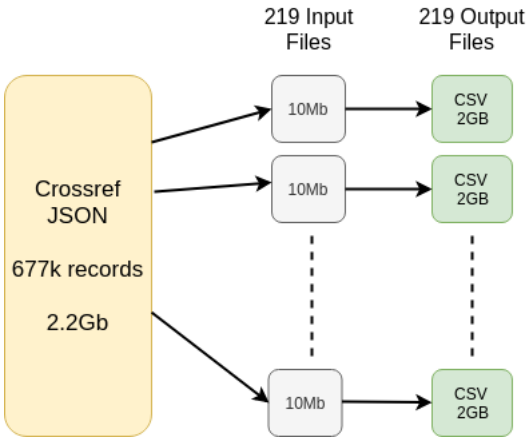

Figure 4.8: The 2.2Gb Crossref JSON file was split into 219 10Mb Input Files. Each input file produced a 2Gb csv output file.

4.2.5 CSL Processor

Citeproc-js [88] 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, integrated into both Mendeley [85] and Zotero [44]. Furthermore, it has an active and responsive community of developers.

In order to use citeproc-js the input data must be in JSON form and follow the citeproc-js JSON schema. The citeproc-js JSON schema requires both an *id* and a *type* field. *Id* is required to uniquely identify the citation and *type* identifies the citation type (e.g. journal, book etc.) The *type* field determines what subsequent fields are allowed.

A number of steps were required to change the JSON data obtained from Crossref into the citeproc-js JSON format. Firstly, Table 4.2 shows tag names which were directly changed. Secondly, empty tags were removed and finally, any tags not allowed as input for citeproc-js were removed.

Crossref Tag	citeproc-js Tag
short-container-title	container-title-short
short-title	shortTitle
journal-article	article-journal
book-chapter	chapter

Table 4.2: Tag names which were changed when making the Crossref JSON compatible with the citeproc-js JSON schema

Figure 4.9 shows the following citation:

Tkaczyk, D., Collins, A., Sheridan, P., & Beel, J. (2018, May). 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.

in Crossref JSON. The same citation is shown in citeproc-js JSON in Figure 4.10. It can be seen that the citeproc JSON is considerably smaller than the Crossref JSON.

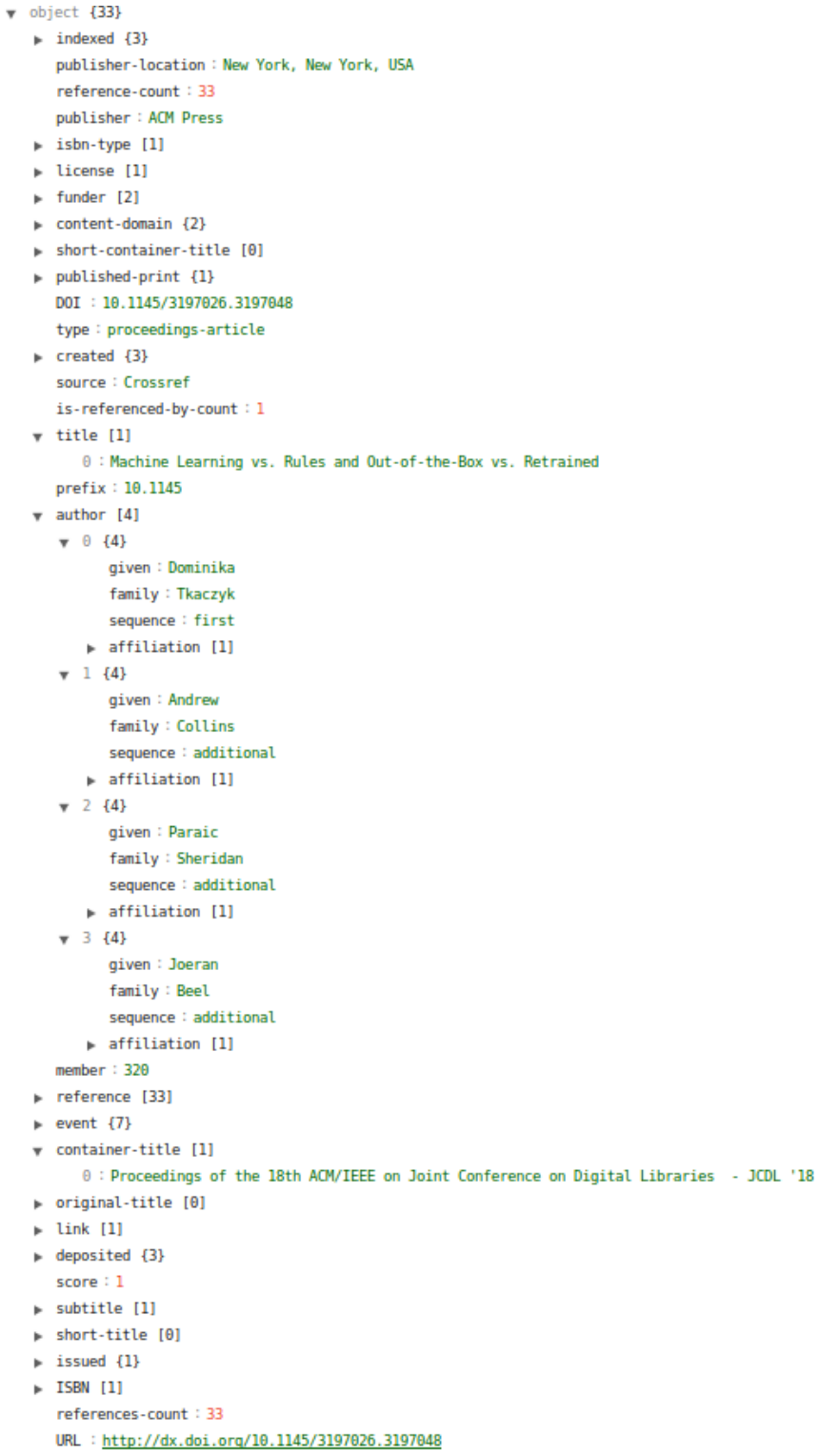

Figure 4.9: An example of JSON returned by Crossref.

```
▼ 0 {11}
    id  : 0
    publisher : ACM Press
    DOI : 10.1145/3197026.3197048
    type : proceedings-article
    source : Crossref
    title : Machine Learning vs. Rules and Out-of-the-Box vs. Retrained
  ▼ author [4]
    ▼ 0 {4}
        given : Dominika
        family : Tkaczyk
        sequence : first
      ▶ affiliation [1]
    ▼ 1 {4}
        given : Andrew
        family : Collins
        sequence : additional
      ▶ affiliation [1]
    ▼ 2 {4}
        given : Paraic
        family : Sheridan
        sequence : additional
      ▶ affiliation [1]
    ▼ 3 {4}
        given : Joeran
        family : Beel
        sequence : additional
      ▶ affiliation [1]
    container-title : Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries  - JCDL '18
  ▶ issued {1}
    ISBN : 9781450351782
    URL : http://dx.doi.org/10.1145/3197026.3197048
```


Figure 4.10: An example of Crossref JSON after being edited for use with citeproc-js.

4.2.6 Indexes

In an effort to provide information for future users of the 1 Billion Dataset 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 (Harvard, MLA etc.)

Both the citation type and the citation style were included as indexes in an effort to reduce space. A separate lookup table is provided for both. Table 4.3 shows the most common citation types along with their respective indexes. The complete list of citation types along with their respective indexes is given in Appendix 10.3.

Index	Citation Type
3	article-journal
5	book
7	chapter
8	dataset
22	paper-conference
27	report
36	reference-entry
37	journal-issue

Table 4.3: Most common citation types in the 1 Billion Dataset and their respective indexes.

4.3 Training existing ML citation parsing tools on 1 Billion Citation Dataset

4.3.1 Introduction

The primary research question I want to address is whether the synthetic 1 Billion Citation Dataset has the potential to be used for training a deep-learning citation parser.

In an effort to address this question the citation parser GROBID is trained using a subset of the 1 Billion Dataset. This model is then compared with the out-of-the-box version of GROBID which was trained using hand-labelled datasets. Further details on the reasoning behind choosing GROBID and the size and make-up of the training data are given in the following sections.

4.3.2 Training GROBID with the 1 Billion Citation Dataset

There were a number of reasons why GROBID was chosen as the citation parsing tool for training using the 1 Billion Dataset. Firstly, as mentioned previously, in a 2018 comparison survey of existing citation parsing tools, Tkaczyk et al. found that GROBID was the state-of-the-art with the highest F1 score of 0.89 [1]. This was a 7.2% improvement on the second highest scoring tool, CERMINE, which had an F1 score of 0.83 [1].

Secondly, as well as being state-of-the-art, GROBID is established, widely-used and open-source. It has been in development since 2008 and been open-source since 2011 [91]. It also has an active and generous community of developers.

Finally, GROBID has excellent documentation. This covers important areas such as the required format of training data, how to carry out training and how to use your own evaluation data. For all of these reasons GROBID

was chosen as the ML citation parsing tool for training on the 1 Billion Dataset.

What follows is a description of the steps taken to make the 1 Billion Dataset compatible for training GROBID, the training datasets created to investigate each of the research questions and the evaluation datasets used.

4.3.3 Making the 1 Billion Citation Dataset compatible with GROBID

The training data for ML citation parsers is typically labelled XML however, the exact make-up of the labels for different citation parsing tools can vary. As outlined in Section 4.2.2, these labels can vary in how fine-grained they are, for example, whether they label individual authors separately or together. They can also vary in whether they differentiate between article titles and book-titles. Finally, the exact labelling can also vary. For example, is the author field labelled as <author> or <name>.

The 1 Billion Dataset was created with as fine-grained labelling as possible. As outlined in Section 4.2.2 an author's first-name, middle name and/or surname were tagged separately. This will enable the 1 Billion Dataset to be used with multiple citation parsing tools regardless of their format. However, this did mean that in it's original format the 1 Billion Dataset was not compatible with GROBID.

A number of steps were taken to make the 1 Billion Dataset compatible with GROBID. Table 4.4 shows the original XML tags in the 1 Billion Dataset that were either unchanged or directly substituted for a different label. For example, the <editor> tag remains exactly as is whilst the <page> tag is changed to <biblScope unit=”page”>.

Original 1 Billion Dataset XML tags	GROBID XML tags
author	author
issued	date
page	biblScope unit="page"
volume	biblScope unit="volume"
issue	biblScope unit="issue"
orgName	orgName
publisher	publisher
editor translator	editor
editor	editor
DOI	idno type="doi"
URL	ptr type="web"

Table 4.4: The original XML tags in the 1 Billion Dataset and the corresponding tag used to train GROBID.

GROBID requires that brackets are placed outside of the date tag. For example, (<date>1984</date>) is acceptable whereas <date>(1984)</date> is not. Entries in the 1 Billion Dataset containing brackets inside the date tag were changed so that the brackets were placed directly outside the date tag.

GROBID labels all the authors as a single field so in order to make the 1 Billion Dataset compatible with GROBID the <family> and <given> tags were removed. The <author> tag remained unchanged. For example, the author field:

<author><family>Grennan</family>,
<given>Mark</given></author>

would appear after editing as:

<author>Grennan, Mark</author>

GROBID has four different types of title tags depending on the nature of the item being referenced. Table 4.5 summarises these tags and where they are used.

GROBID Title Tag	Description
title level="a"	For article title
title level="j"	For journal title
title level="s"	For series title (e.g. Lecture Notes in Mathematics)
title level="m"	For book title and any other non-journal bibliographic item (e.g. conference proceedings title)

Table 4.5: The four different title tags used by GROBID along with a description of where they are used.

The 1 Billion Dataset has two different types of title tags, namely <title> and <container-title>. The <title> tag is used to label the article or chapter title of a reference and maps directly to GROBID's <title level="a"> tag. However, <container-title> is used to label journals, books and non-bibliographic titles. No distinction is made between them.

A challenge was how to map the <container-title> tag to GROBID's three other different title tags. This challenge was addressed by using the citation type lookup index as given in Table 4.3. The citation type index details the type of citation (e.g. book, journal-article, chapter etc.). Based on this information, an appropriate tag could be chosen as a substitute for <container-title>. The mappings chosen for the most common citation types are given in Table 4.6.

Citation Type	GROBID Title Tag
article-journal	title level=”j”
book	title level=”m”
chapter	title level=”m”
paper-conference	title level=”m”
dataset	title level=”m”

Table 4.6: The title tag used for each citation type

4.3.4 Training Datasets

A citation parser is typically used in information retrieval by academic search engines and digital libraries. Given that the aim of the citation parser is to perform well on unseen citation strings, ideally a training dataset would reflect the data on which these searches are carried out and the distribution of citation types, styles and domains in popular use. Unfortunately we can only estimate this data so the choice was made to experiment with different make-ups of training data.

Figure 4.11 shows the distribution of citation types of three different training datasets used. All three training datasets were created using the 1 Billion Dataset and each dataset contains over 1,000 citation styles.

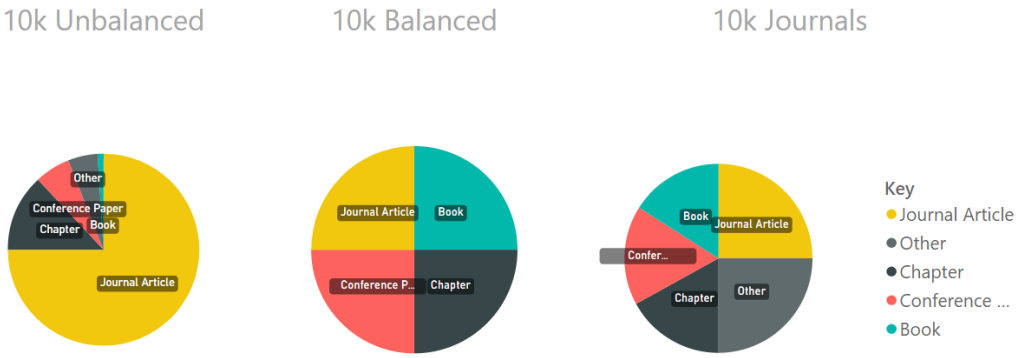

Figure 4.11: The distribution of citation types in three different training datasets used to train GROBID.

“10k Unbalanced” contains 75% journal articles, 13% chapters, 6% conference papers and 1% books. “10k Balanced” contains 25% of each type: journal article, chapter, conference paper and books. “10k Journals” contains 25% journal articles, 17% chapters, 17% conference papers and 16% books. It also contains 25% citations which just contain a journal title and not a paper title.

In an effort to investigate how the size of the training dataset affects the performance of GROBID six different training datasets were used. These had sizes: 1k, 3k, 5k, 10k, 20k and 40k.

Finally, as outlined in Section 3.3.2, different citation parsing tools have more or less fine-grained labelling. Here, the out-of-the-box GROBID training data is more fine-grained than the 1 Billion Dataset. The following tags are present in the out-of-the-box training data and not in the 1 Billion Dataset:

- institution: the institution for theses or technical reports
- note: a description of the type of technical report (E.g. PhD, MSc etc.) or any indications related to the reference not covered by a previous tag [1]
- pubPlace: location of the publishing institution

In an effort to determine whether there is any difference, aside from granularity, between models trained with hand-labelled data and the synthetic 1 Billion Dataset it is necessary to compare models trained with the same granularity of data. Therefore, GROBID was retrained using it's out-of-the-box training data with citations containing the following tags removed: institution, note and pubPlace. This model was compared with GROBID retrained on the 1 Billion Dataset.

4.4 Evaluation Datasets

The challenges associated with evaluation datasets currently in use were extensively discussed in Section 3.3.3 of Related Work. Briefly recapping the main points, there is no gold-standard evaluation dataset and the datasets that do exist are small and homogeneous, often coming from a single domain.

Further, it was found that some of the existing data available was used to train the out-of-the-box version of GROBID. Using this same data for evaluation purposes would lead to heavily biased and meaningless results.

In order to draw fair conclusions there was a need for a new, "unseen" evaluation dataset. Therefore, both in an effort to fairly evaluate the 1 Billion Dataset and as a contribution to the community, time was devoted to creating a completely new evaluation dataset.

For ease of discussion, the out-of-the-box version of GROBID which was trained using hand-labelled data will be referred to as the "hand-labelled" model. GROBID retrained on a subset (10,000 citations) of the 1 Billion Dataset will be referred to as the "synthetic" model.

[1] *note* is contained within the 1 Billion Dataset but the information contained within the note tags of the 1 Billion Dataset and what GROBID expects do not match up therefore this field was excluded.

4.4.1 Unseen Evaluation Dataset

An evaluation dataset, although much smaller than the 1 Billion Dataset, should still be diverse in citation style, language and domain. Equally it should contain errors commonly found in citation strings: OCR errors, spelling errors, incorrect spacing, punctuation etc. What follows is a brief description of the steps taken to create this dataset.

First, twenty words were chosen from the homepage of different university departments. The words were: bone, recommender systems, running, war, crop, monetary, migration, imprisonment, hubble, obstetrics, photonics, carbon, cellulose, evolutionary, resolutionary, paleobiology, penal, leadership, soil, musicology.

Each of the twenty words was then searched in Google Scholar. From each of these searches the first four available PDFs were taken and within each PDF four citation strings were chosen at random. This gave approximately sixteen citation strings for each of the twenty keywords and in total, there were 300 citation strings.

GROBID's out-of-the-box model was used to preliminarily tag each of these 300 citation strings. Following this, each string was manually checked and edited to ensure that it had been correctly tagged. Errors included within the dataset include: OCR errors, incorrect spacing and punctuation.

Both the hand-labelled and synthetic models were then evaluated using this "unseen" evaluation dataset.

4.4.2 CORA Dataset

Both models, hand-labelled and synthetic, were also evaluated using the popular CORA dataset. This dataset was obtained in it's original format from the website of Andrew McCallum [70], the creator of the dataset.

The limitations of the CORA dataset were extensively discussed in Section 3.3.3. Furthermore, it was found on investigation that some of the CORA dataset was included in the out-of-the-box GROBID training dataset. For this reason, CORA was not used to compare the relative performance of the hand-labelled and synthetic models.

However there were two reasons for evaluating the synthetic model on CORA. Firstly, it was included for completeness. In this way the model trained on synthetic data could easily be compared with other existing tools that had been evaluated on CORA. Secondly, it was used to highlight any potential areas of weakness in the 1 Billion Dataset.

In spite of it's popularity, reformatting the CORA dataset to be compatible with GROBID was not without challenges. Similar to how the 1 Billion Dataset had to be re-formatted to be compatible with GROBID some of the original tags in the CORA dataset had to be renamed. A complete list of the mappings from the original tags to the reformatted tags is given in Table4.7.

Original CORA tags	GROBID tags
author	author
title	title level="a"
journal	title level="j"
volume	biblScope unit="volume"
pages	biblScope unit="page"
address	pubPlace
booktitle	title level="m"
date	date
publisher	publisher
editor	editor
institution	institution
note	note
year	date

Table 4.7: The original XML CORA tags and the corresponding tag used to train GROBID.

In many cases, in it's original format, a space existed both before a closing tag and after an opening tag. For example, the author field often looked like:

<author> Mark Grennan </author>

as opposed to the following, without spaces:

<author>Mark Grennan</author>

This was a problem as GROBID is sensitive to blank spaces. Therefore, had GROBID tagged the author field as: <author>Mark Grennan</author> this would be marked as incorrect. In order to rectify this, any spaces occurring either after an opening tag or before a closing tag were removed.

Similarly, GROBID is sensitive to punctuation. A number of fields, such as volume and pages, were changed so that punctuation was placed outside the tag. It is important to note that moving the punctuation outside of the tag did not change the structure of the citation string but rather, how the string was labelled. Examples of punctuation that were changed are given in Table 4.8.

As well as this, any brackets that occurred within the date tag were moved outside the tag. Further, 47 duplicate citations were identified and removed.

Finally, XML by it's nature is unforgiving. Any misspelt tags or missing tags will prevent the document from being processed. In the original CORA dataset a number of citations had these errors. This added to the time it took to reformat CORA.

Original CORA tags	Reformatted CORA tags
<pages>893,</pages>	<pages>893</pages>,
<volume> 76, </volume>	<volume>76</volume>,
<date> 1981. </date>	<date>1981</date>.
<volume> 31: </volume>	<volume>31</volume>:
<pages> pp. 141-155. </pages>	pp. <pages>141-155</pages>.

Table 4.8: Examples of punctuation that were moved outside of the tags in the CORA dataset.

4.4.3 Other Evaluation Datasets

As well as evaluating on the "unseen" dataset and the CORA dataset, both models were also evaluated using a subset of the 1 Billion Dataset (5,000 citations).

Further, GROBID was also re-trained using 70% of it's original training data (5,460 citations). The remaining 30% of it's original training data (2,340 citations) was then used as an evaluation dataset. Here, GROBID retrained on 70% of it's original data was compared with the synthetic model.

Both of these evaluation datasets are heavily biased and the results should not be used to draw conclusions about the relative performance of the synthetic and hand-labelled training data. These evaluations were primarily a form of exploratory analysis.

The expectation was that the synthetic model would perform better when evaluated on the synthetic 1 Billion Dataset and similarly, that the hand-labelled model would perform better when evaluated on the hand-labelled 30% dataset.

4.5 Evaluation Metrics

As discussed in Section 3.3.1 the most commonly reported metric for evaluating citation parsing is F1 although there is no single accepted metric. For completeness, recall, precision and F1 are reported. Formulas and details about the calculation of these metrics can be found in Section 2.3 of the Background Chapter.

Both micro and macro-average results are reported at the field-level. Again, details about how these averages are calculated can be found in Section 2.3 of the Background Chapter.

Both averages are reported at the field-level. This was deemed to be more meaningful than the token-level results. Finally, as well as the overall average, recall, precision and F1 are reported for the following ten fields: author, title, book-title, journal, date, editor, volume, issue, pages and publisher.

Chapter 5

Results

5.1 Analysis of 1 Billion Citation Dataset

The final format of the 1 Billion Citation Dataset is a csv with four columns: DOI, citation type, citation style and labelled citation string. A separate lookup table is provided for citation type and citation style. Table 5.1 gives an example of the layout.

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

Table 5.1: The final format of the 1 Billion Dataset csv. Columns exist for DOI, citation type, citation style and XML labelled citation. A separate lookup table is provided for the indexes for citation type and style.

The final dataset has 991,411,100 labelled citation strings and is 438Gb in size. This is composed of 219 * 2Gb files. Figure 5.1 gives the percentage breakdown of the 1 Billion Dataset by citation type. Journal articles are the most common type of citation making up 75.9% of the 1 Billion Dataset, followed by chapter citations (12.4%) and conference papers (5.6%).

Table 5.2 provides further detail with columns included for total number of labelled citations, number of unique citation strings and percentage of dataset. The number of unique citations is found by dividing the total number of citations by the 1,564 citation styles. The total number of unique citations is 633,895 which means that of the original 677,00 Crossref records, 43,105 were not processed correctly. A complete breakdown of all categories including "other" is given in Appendix 10.2.

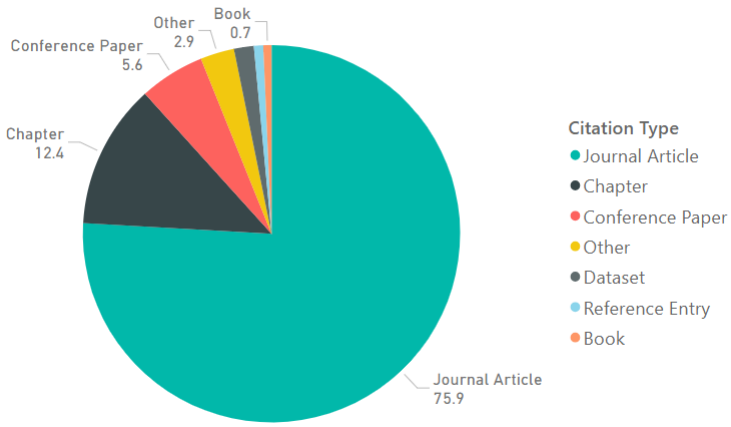

Figure 5.1: The percentage breakdown of citation types contained within the 1 Billion Dataset

Citation Type	Number of Labelled Citation Strings	Number of 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%**

Table 5.2: Breakdown of Citation Types contained within 1 Billion Dataset.

A look at the percentage breakdown of citation styles reveals that of the 1,566 original styles, four did not work and 55 did not process every citation correctly. This largely explains the missing records.

Each of the 1,505 citation styles which processed every record correctly contains 633,966 unique citations. The total number of citation types in each citation style is given in Table 5.3.

Citation Type	Number of Labelled Citation Strings	Percentage
Journal Article	481,203	75.9%
Chapter	78,616	12.4%
Conference Paper	35,504	5,6%
Dataset	10,778	1.7%
Reference Entry	5,072	0.8%
Book	4,438	0.7%
Other	18,386	2.9%
Total	**633,996**	**100%**

Table 5.3: Breakdown of Citation Types contained within each Citation Style

5.2 Evaluating on the 1 Billion Dataset

An evaluation dataset composed of 5,000 synthetic citations was created from the 1 Billion Dataset. The results of evaluating the synthetic and hand-labelled training datasets using this synthetic evaluation dataset are given in Table 5.4. Here, micro-average for recall, precision and F1 are reported at the field-level.

As expected, the model trained using synthetic training data was superior to the model trained using hand-labelled data when evaluating on synthetic data. There was a 32.9% difference between the models trained using synthetic and hand-labelled data (F1 0.93 vs F1 0.70, $p < 0.01$). As discussed in Section 4.4.3 these results are heavily biased.

Training Data	Precision	Recall	F1
Hand-Labelled	0.70	0.71	0.70
Synthetic	**0.93**	**0.93**	**0.93**

Table 5.4: Evaluating models trained using hand-labelled and synthetic training data using a synthetic evaluation dataset created from the 1 Billion Dataset.

Table 5.5 shows the F1 broken down by field for the models trained using hand-labelled and synthetic training data. The fields in which there is the largest difference in F1 are: book-title (F1 0.29 vs F1 0.75) , publisher (F1 0.55 vs F1 0.94) and title (F1 0.54 vs F1 0.87).

Field	Hand-Labelled Training Data	Synthetic Training Data
author	0.74	**0.93**
book-title	0.29	**0.75**
date	0.92	**0.96**
issue	0.81	**0.91**
journal	0.63	**0.90**
pages	0.90	**0.98**
publisher	0.55	**0.94**
title	0.54	**0.87**
volume	0.88	**0.94**
Micro-Average	0.70	**0.93**
Macro-Average	0.69	**0.91**

Table 5.5: F1 broken down by field for the models trained using hand-labelled and synthetic training data. Evaluated on 5,000 synthetic citations from the 1 Billion Dataset.

5.3 Evaluating on 30% of GROBID's Original Training Data

GROBID was trained using 70% (5,460 hand-labelled citations) of it's original training data. The remaining 30% (2,340 hand-labelled citations) was then used as an evaluation dataset.

The results of evaluating the synthetic and hand-labelled training datasets using this evaluation dataset are given in Table 5.6. Again, micro-averages for recall, precision and F1 are reported at the field-level.

Training Data	Precision	Recall	F1
Hand-Labelled (70% original training data)	**0.95**	**0.95**	**0.95**
Synthetic	0.83	0.75	0.78

Table 5.6: Evaluating models trained using hand-labelled and synthetic training data using an evaluation dataset created from 30% of GROBID's original training data.

This time, as expected, the results have been reversed. When evaluating using 30% of GROBID's original training data the model trained using hand-labelled data outperforms the model trained with the synthetic data. Again, it is worth acknowledging that these results are also biased.

The F1 values broken down by field are given in Table 5.7. The fields with the largest difference in F1 are: book-title (F1 0.77 vs 0.15) and publisher (F1 0.87 vs F1 0.38).

Field	Hand-Labelled Training Data	Synthetic Training Data
author	**0.95**	0.83
book-title	**0.77**	0.15
date	**0.99**	0.95
issue	**0.91**	0.73
journal	**0.95**	0.78
pages	**0.98**	0.78
publisher	**0.87**	0.38
title	**0.93**	0.75
volume	**0.98**	0.86
Micro-Average	**0.95**	0.78
Macro-Average	**0.92**	0.69

Table 5.7: F1 broken down by field for the models trained using hand-labelled and synthetic training data. Evaluated on 2,340 hand-labelled citations from GROBID's original training data.

Although the results have been reversed it is interesting to note that the same fields – book-title and publisher - have the largest difference in both evaluations. This points to an area in which the labelling of the 1 Billion Dataset may differ from the out-of-the-box GROBID training data.

5.4 Evaluating on CORA

As detailed in Section 4.4.2, some of the CORA dataset is contained within the out-of-the-box GROBID training data. Further, the CORA dataset has limitations as outlined in Section 3.3.3. Therefore evaluations using CORA should be seen as a further tool for investigation rather than a measure of relative performance.

The results of evaluating on CORA models trained using hand-labelled and synthetic training data are given in Table 5.8. The percentage difference between the model trained using hand-labelled data and the model trained using synthetic data is larger for recall (19.0%) than for precision (11.1%).

Training Data	Precision	Recall	F1
Hand-Labelled	**0.72**	**0.79**	**0.75**
Synthetic	0.64	0.64	0.64

Table 5.8: Evaluating models trained using hand-labelled and synthetic data on the CORA dataset.

The field-level results are given in Table 5.9. It is noteworthy that the fields with the largest difference in F1 are again: book-title (F1 0.67 vs F1 0.16), publisher (F1 0.83 vs F1 0.33) and title (F1 0.90 vs F1 0.79).

Field	Hand-Labelled Training Data	Synthetic Training Data
author	**0.91**	0.89
book-title	**0.67**	0.16
date	**0.87**	0.86
issue	**0.64**	0.60
journal	0.52	**0.53**
pages	**0.75**	0.73
publisher	**0.83**	0.33
title	**0.90**	0.79
volume	**0.76**	0.70
Micro-Average	**0.75**	0.64
Macro-Average	**0.76**	0.62

Table 5.9: F1 values broken down by field when evaluating on the CORA dataset.

The difference in results for the fields: author, date, issue, journal, pages and volume are not significant at the 5% significance level.

5.5 Evaluating on the "Unseen" Dataset

The results of evaluating on the "unseen" dataset models trained using hand-labelled and synthetic training data are given in Table 5.10. The model trained with hand-labelled data outperforms the model trained with synthetic data in precision, recall and F1. The percentage difference between the two models is again larger for recall (12.8%) than for precision (6.8%).

Training Data	Precision	Recall	F1
Hand-Labelled	**0.89**	**0.87**	**0.88**
Synthetic	0.83	0.76	0.80

Table 5.10: Evaluating models trained using hand-labelled and synthetic data on the "unseen" dataset.

The field-level results are given in Table 5.11. Although the model trained using hand-labelled data has a higher F1 across all fields the results are not significant at the 5% significance level for the fields: date, issue, pages and volume. The fields with the most significant differences are book-title ($p < 0.005$) and publisher ($p < 0.00001$).

Field	Hand-Labelled Training Data	Synthetic Training Data
author	**0.91**	0.84
book-title	**0.35**	0.16
date	**0.97**	**0.97**
issue	**0.87**	0.83
journal	**0.86**	0.79
pages	**0.96**	0.94
publisher	**0.87**	0.46
title	**0.79**	0.70
volume	**0.96**	0.94
Micro-Average	**0.88**	0.80
Macro-Average	**0.84**	0.74

Table 5.11: F1 value broken down by field when evaluating on the "unseen" dataset.

5.6 Effect of granularity

GROBID was trained using it's original training data with citations containing the following tags removed: *institution, note* and *pubPlace*. The F1 scores for this model along with the model trained using the synthetic data are detailed in Table 5.12.

Both models were evaluated on the "unseen" Dataset. Although the model trained using hand-labelled data has a higher micro-average F1 0.82 than the model trained using the synthetic 1 Billion Dataset F1 0.80, the results are not significant at 5% level.

Training Data	Precision	Recall	F1
Hand-labelled	**0.84**	**0.79**	**0.82**
Synthetic	0.83	0.76	0.80

Table 5.12: Evaluating using the "unseen" dataset when both training datasets have the same level of granularity. The difference in F1 (0.82 vs 0.80) is not significant at the 5% significance level.

Table 5.13 shows the results at the field level for both models. Here, it can be seen that the macro-average F1 0.74 is the same for both models. The out-of-the-box model has a higher F1 for author, issue and title whilst the model trained using synthetic data has a higher F1 for book-title and publisher.

Removing citations containing the *pubPlace* tag from GROBID's original training data has significantly reduced the performance of GROBID in predicting book-title. This is further discussed in Chapter 6 on Discussion.

Field	Hand-Labelled Training Data	Synthetic Training Data
author	**0.89**	0.84
book-title	0.15	**0.16**
date	**0.98**	0.97
issue	**0.87**	0.83
journal	**0.81**	0.79
pages	**0.96**	0.94
publisher	0.27	**0.46**
title	**0.76**	0.70
volume	**0.95**	0.94
Micro-Average	**0.82**	0.80
Macro-Average	**0.74**	**0.74**

Table 5.13: F1 values broken down by field when both training datasets have the same level of granularity.

5.7 Effect of changing make-up of Training Data

As outlined in Section 3.4, three different datasets were created to examine the effect of changing the distribution of citation types within a training dataset. These were named “10k Unbalanced”, “10k Balanced” and “10k Journals”.

Figure 5.2 shows the results for precision, recall and F1 for models trained on each of the three datasets. The models were evaluated on the “Unseen” dataset. The model trained on “10k Unbalanced” dataset has the highest precision 0.76, recall 0.83 and F1 0.80. The difference between the model trained on the “10k Unbalanced” dataset and the other two models is significant at 5% significance level.

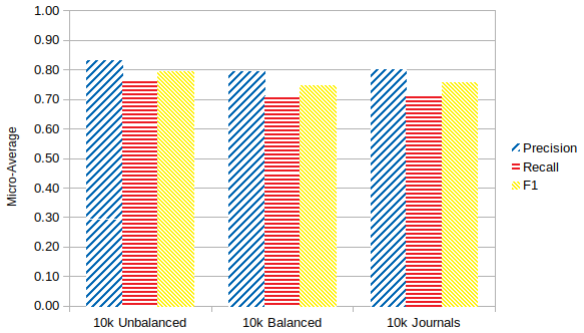

Figure 5.2: Precision, recall and F1 for models trained on the "10k Balanced", "10k Unbalanced" and "10k Journals" datasets. Each dataset was created from the 1 Billion Dataset.

Table 5.14 shows the results broken down by field. Interestingly, although “10k Balanced” contains more book citations (25%) compared with “10k Unbalanced” (1%) it does worse at predicting the book-title field (F1 0.12 vs F1 0.16) and the title field (F1 0.56 vs F1 0.70). Adding in citations containing just the journal title, “10k Journals”, did not improve the result for predicting journal title (“10k Unbalanced” F1 0.79 vs “10k Journals” F1 0.72).

Field	10k Unbalanced	10k Balanced	10k Journals
author	**0.84**	0.80	0.80
book-title	**0.16**	0.12	0.10
date	**0.97**	0.94	**0.97**
issue	**0.83**	0.77	0.81
journal	**0.79**	0.68	0.72
pages	0.94	0.92	**0.95**
publisher	**0.46**	**0.46**	0.40
title	**0.70**	0.56	0.56
volume	**0.94**	0.92	0.93
Micro-Average	**0.80**	0.75	0.76
Macro-Average	**0.74**	0.69	0.69

Table 5.14: F1 broken down by field for GROBID trained on "10k Unbalanced", "10k Balanced" and "10k Journals". Evaluation was conducted on the "unseen" dataset.

5.8 Effect of increasing the size of the training dataset

Figure 5.3 shows the F1 field-level micro-average value as the size of the training dataset increases. Results are shown for each of the four evaluation datasets: 1 Billion 5k Sample, 30% Grobid Evaluation Dataset, CORA and the “Unseen” dataset.

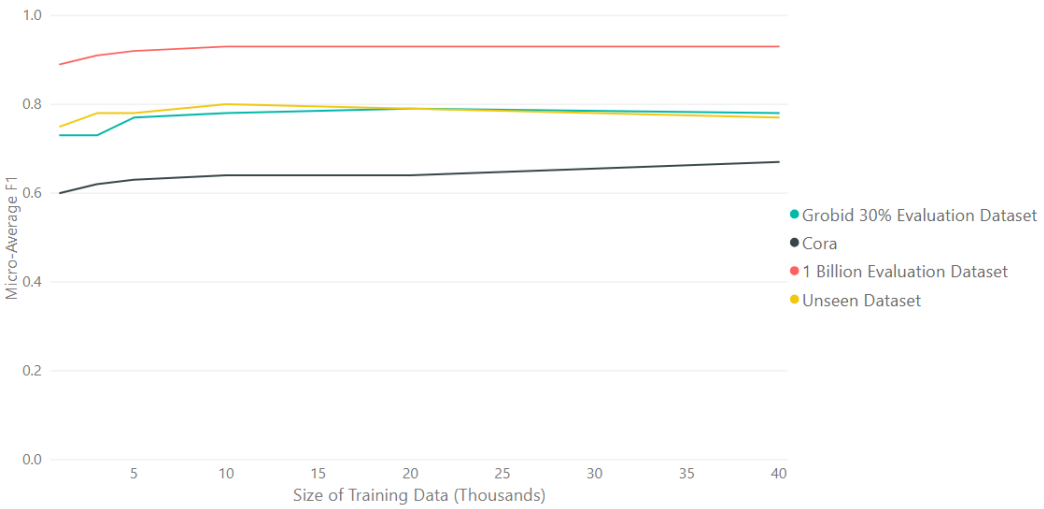

Figure 5.3: The F1 score for each evaluation dataset as the size of the training dataset increases

The results show that increasing the size of the training data from 1,000 to 10,000 improved F1 regardless of the evaluation dataset. However, when the training dataset increased in size from 10,000 to 40,0000 there was no clear pattern. The F1 score either didn't improve further (1 Billion 5k Sample, 30% Grobid Evaluation Dataset), regressed ("Unseen" dataset) or improved (CORA). The overall indication is that increasing the size of the training dataset above 10,000 does not lead to a clear improvement in performance.

Figure 5.4 shows the length of time it took to train the GROBID model as the size of the training dataset increased. The training time increased linearly as the size of the training data increased. A model with a training dataset of size 40k took 76 hours to train. Training was carried out using n1-standard-8 CPUs on Google Compute Engine.

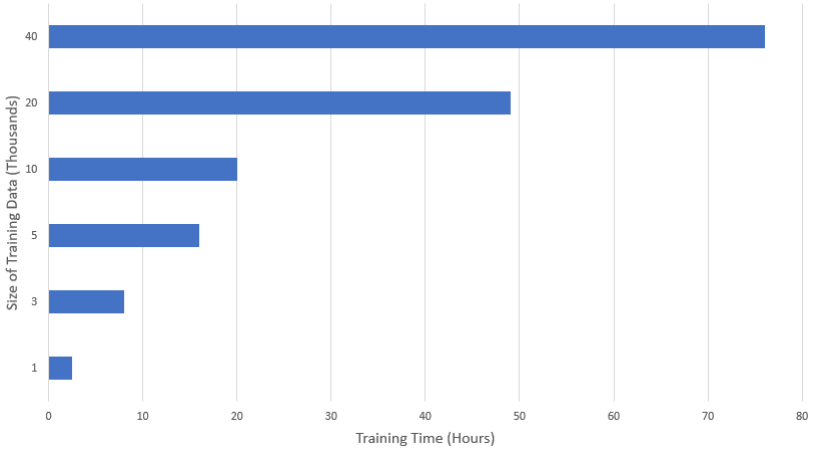

Figure 5.4: The training time in hours as the size of the training dataset increases. Training was carried out using n1-standard-8 CPUs.

Chapter 6

Discussion

6.1 Analysis of the 1 Billion Citation Dataset

The requirements for the 1 Billion Dataset as outlined in the Research Aims in Section 1.4 were that it should be diverse in citation styles, type and domain and also, that it should be large. With over 991k labelled citations the final dataset satisfies the requirement to be large.

Secondly, the dataset is diverse in citation styles with a total of 1,564 citation styles represented. Each citation style has over 633k labelled citations. An advantage of both the size and diversity of the dataset is that should a future researcher wish to just focus on a particular style or styles there should be adequate labelled citations available.

The third requirement for the dataset was that it is diverse in citation type. At first glance, with 75.9% of the dataset made up of journal article citations, 12.9% made up of chapter citations and all other citation types making up the remaining 11.2%, it appears that the dataset does not capture this required diversity.

However, an advantage of the size of the 1 Billion Dataset is that even for less common citation types there still remains a large number of citations. For example, monograph makes up only 0.4% of the dataset but this includes nearly 4 million citations. Similarly, dissertation makes up 0.3% of the dataset but contains over 2.5 million citations. A potential negative is the low proportion of books (0.7%). However, in spite of the low proportion there still remains over 7 million labelled book citations.

Finally, the source code and accompanying documentation is freely available on github (https://github.com/BeelGroup/1-Billion-Citation-Dataset) allowing anyone to create their own dataset containing citation metdata, styles and types as required.

6.2 Format of the 1 Billion Citation Dataset

The final format of the 1 Billion Dataset is a csv with four columns: DOI, citation type, citation style and labelled citation string. An advantage of this format and having a citation type lookup column means that it is easy to create a dataset composed of any desired citation types. A future researcher would be easily able to supplement training data that they may already have with other desired citation types from the 1 Billion Dataset.

A challenge associated with training ML citation parsing tools is that many tools have different formats for their training data. As discussed in the Methodology in Section 4.2.2, different tools may have more or less fine-grained training data. They may also tag field names differently. For example, some tools use <author> whilst others use <name>.

When creating the 1 Billion Dataset it was decided to tag individual authors' names independently. For example, *Mark Grennan* is tagged as:

<author><given>Mark</given> <family>Grennan</family></author>

An advantage of this format is that regardless of whether a citation parsing tool labels authors as a single field, as individual authors or as individual names, the 1 Billion Dataset can be made compatible with each of these formats. For example, to create a single author field simply involves removing the <given> and <family> tags.

A potential negative of the format of the 1 Billion Dataset is the use of the <container-title> tag. This tag is used interchangeably to represent journal titles, book titles and series titles. A disadvantage here is that some citation parsing tools may have different tags for each of these items. For example, <journal> may be used for a journal title and <book> for a book title.

In Section 4.3.3 it was outlined how the citation type, given in the citation type index, of a particular citation could be used to map the <container-title> tag to an appropriate label. For example, if the citation type is book the <container-title> tag can be changed to <book>. However, this increases the labour associated with using the dataset.

An advantage of the work carried out during this project is that the code associated with converting tags from one label to another (e.g. <author> to <name>) has already been developed. This work is detailed on the project's github. Aside from the <container-title> tag, converting from one label to another using this code simply amounts to changing entries in a dictionary. Time was devoted to ensuring that the documentation associated with this work is clear.

6.3 Addressing the Research Question

The principle research question being addressed in this project is what is the potential of the 1 Billion Dataset for training a deep-learning citation parsing model.

Evaluating on the "unseen" dataset the results showed that the model trained on hand-labelled data was superior to the model trained using synthetic data (Hand-Labelled F1 0.88 vs Synthetic F1 0.80). However, once the training data had the same level of granularity - *pubPlace, note* and *institution* tags were removed - it was found that there was no significant difference between the two models (Hand-Labelled F1 0.82 vs Synthetic F1 0.80).

Why did granularity (*pubPlace, note, institution*) make such a big difference? I believe that a lot of the observed difference can be explained by the lack of a book-title tag. GROBID requires that all non-journal bibliographic items use the same title tag. For example, titles for dissertations, conference papers and books are all labelled the same in the training data.

However, the hand-labelled training data has an additional *pubPlace* tag. This is used to tag publishing location. The advantage of this tag, I believe, is that it is primarily used with books and therefore, essentially amounts to a *book* tag. This enables the model with *pubPlace* tags to superiorly learn what is, and what is not, a book citation.

Without access to this labelling the model trained on synthetic data has no way of differentiating between different non-journal items. Titles for thesis, conference papers and books were all labelled the same in the training data and yet, the model was being asked to differentiate between *titles* and *book-titles*.

It is important to note that although *pubPlace* is not a tag within the 1 Billion Dataset, as outlined previously, it is possible to change the container-tag title within the 1 Billion Dataset to appropriate journal, book, dissertation etc. tags as required. Therefore the lack of granularity for non-journal titles is not an inherent flaw of the 1 Billion Dataset but merely the way GROBID requires it's training data to be formatted.

The other field which the model trained on synthetic data consistently struggled to predict was *publisher*. This field is also associated with book citations and again, I think the lack of a separate book tag, or suitable synonym such as *pubPlace*, is the reason for this poor predictive performance.

Further, *journal title* and *book-title* are inherently linked. If the model incorrectly predicts *journal title* for a *book-title* the precision of the *journal title* field will be negatively affected as will the recall for the *book-title* field. Therefore, if the model trained on hand-labelled data is superior in predicting *book-title* this will result in better results for all related fields including: journal, title and publisher.

Fields not linked to title such as *volume, issue, page number* and *date*

have similar results in both models (Hand-labelled Macro-Average F1 0.94 vs Synthetic Macro-Average F1 0.92). This provides further evidence that there is no significant difference between the performance of the model trained on synthetic data compared with the model trained on hand-labelled data.

The results showed that for Conditional Random Fields, the algorithm used by GROBID, there is no clear improvement in performance once the size of the training dataset goes above 10k. Ultimately, the dataset was designed to enable the application of deep-learning in citation parsing and typically, the performance of deep-learning models improves with more data. It is unlikely that these results would be replicated with a deep-learning model.

That said, the flattening of performance as the size of the training data increases may point to a potential limitation of the 1 Billion Dataset. The dataset was created without noise. There is no extra white spaces, OCR, spelling or punctuation errors within the dataset. It is possible that above a certain size this lack of noise reduces how much the model can learn from the training data.

The final research question addressed was how the make-up of the training dataset affects the performance of the citation parser. The results here indicated that the distribution of citation types has a significant impact on overall performance.

The results also showed that predicting the performance from the make up of the training data was not always intuitive. For example, the "10k Balanced" dataset contained 25% book citations yet performed worse in predicting book-titles than the "10k Unbalanced" dataset, which only contained 1% books. Here, the results indicated that it is best to experiment with different distributions of citation types in the training data when trying to improve performance.

Chapter 7

Conclusion

As detailed in the Related Work in Chapter 3 the majority of existing citation parsing tools use small, hand-labelled training datasets. These are typically in the region of five to ten thousand labelled citation strings.

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. [32] and Prasad et al. [13] in 2018 has given an early indication that citation parsing is also likely to benefit from applying deep-learning methods.

However, to date the largest training dataset, used by Rodrigues et al. [32], has only 40,000 labelled citation strings. In the area of deep-learning this would be considered a small dataset. Further, this dataset lacks diversity, with citations coming solely from the humanities.

The 1 Billion Dataset 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. It also contains indexes for citation type and style which enables the creation of datasets to fit any requirement.

When granularity is the same, the 1 Billion Dataset has been shown to be on a par with the state-of-the-art hand-labelled training dataset. This indicates that it should be very suitable for applying to deep-learning methods.

With deep-learning models outperforming more "traditional" models in so many diverse fields, the signs are that training a deep-learning model on the 1 Billion Dataset may lead to significant improvements in the accuracy of citation parsing. The 1 Billion Dataset can enable this future work to develop in the application of deep-learning to citation parsing.

Finally, training a ML citation parsing tool using synthetic data is novel. Typically ML citation parsing tools are trained using hand-labelled data and although Ryan [74] created a synthetic training dataset, there is no account that this dataset was successfully used to train a citation parsing tool. To the best of my knowledge, this is the first time experiments have been con-

ducted using completely synthetic training data.

Below are listed the key contributions of this project:

- The 1 Billion Citation Dataset
  As outlined above, this dataset is unique in the field. It is unique in both scale and diversity. It will be freely available for future researchers to use and has been shown to have the potential to significantly advance the field of citation parsing.

- Code and documentation
  Time and effort was put into both the code and documentation for this project. The work taken in developing code to convert the 1 Billion Dataset from one format to another is likely to be of benefit to any future researcher who wishes to use the 1 Billion Dataset. This work is available here: https://github.com/BeelGroup/1-Billion-Citation-Dataset.

- "Unseen" evaluation dataset
  The "unseen" evaluation dataset contains 300 hand-labelled citations from a diverse range of fields. With the limitations of existing evaluation datasets detailed extensively in Section 3.3.3 in Related Work, this dataset, although small, may be used again as an evaluation dataset in the future. Further, it may help in the development of a gold-standard evaluation dataset.

- Contribution to open-source community
  In working with the existing libraries citeproc-js [88] and GROBID [92] I had two contributions accepted into their latest releases. With GROBID there was a dependency issue. I suggested an upgrade that fixed the issue and this was adopted. (https://github.com/kermitt2/grobid/issues/432) With citeproc-js, I caught a bug associated with an incorrect numeric type for a variable. I suggested a fix and this was adopted in the following release. (https://github.com/Juris-M/citeproc-js/issues/110)

- Summary of existing training and evaluation datasets
  Although the challenges associated with citation parsing have been well documented no previous paper has given a summary of the size and domain of existing training and evaluation datasets. I conducted a survey of 31 papers between 2000 and 2019. From this I summarised all available information related to the size and domain of the training and evaluation datasets used. I also summarised the evaluation metrics used in each paper. I believe that this is a significant contribution to the field and should help to raise awareness of the potential limitations of existing datasets.

Chapter 8

Summary

Chapter 1 introduces the background and motivation for creating the 1 Billion Dataset. It details the research problem, research question and research aims of the project.

Chapter 2 gives some background information including an introduction to citation strings, styles and fields. It also gives an overview of citation parsing as well as detailing common evaluation methods.

Chapter 3 gives a detailed literature review. This includes an in depth study into existing datasets and a discussion of the challenges associated with comparing different approaches within the field.

Chapter 4 details the methodology. This includes a description of Citation Style Languages and how they were adapted to create the 1 Billion Citation Dataset. It also list the motivation for training GROBID using the 1 Billion Dataset along with a description of the training datasets used. Finally, it details the steps taken in creating an "unseen" evaluation dataset.

Chapter 5 lists the results. This includes an analysis of the 1 Billion Citation Dataset as well as the results in training GROBID using a subset of the 1 Billion Dataset. It details what effects the granularity, the make-up and the size of the training data had on performance.

Chapter 6 provides a discussion of the results. This includes the advantages and disadvantages of the format of the 1 Billion Dataset. It also provides context for the results of training GROBID using the 1 Billion Dataset.

Chapter 7 provides a conclusion, summarising the significance of the 1 Billion Citation Dataset and the work completed. It also details the overall key contributions of this dissertation.

Finally, chapter 9 provides suggestions for areas of future work as well as a discussion of the limitations of this project.

Chapter 9

Future Work and Limitations

One of the downsides of this project was that no deep-learning citation parsing model was successfully trained using the 1 Billion Dataset. As previously mentioned, there currently exist only two published accounts of a deep-learning approach to citation parsing, namely Rodrigues et al. [32] and Prasad et al. [13]. Extensive time and effort was given to trying to train these tools using the 1 Billion Dataset but ultimately this proved unsuccessful.

With Rodrigues et al. [32] I ran into dependency issues. At the time of writing there are three open issues on their github and all relate to similar dependency challenges. With Prasad et al. [13] I emailed the author and opened an issue on github however again, I faced implementation blocks. It is perhaps worth noting, that Tkaczyk et al.'s comparison survey of existing citation parsing tools in 2018 also faced similar obstacles and they were not able to include the deep-learning work of Prasad et al. due to "missing resources" [1].

With this in mind, the obvious area for future work is to train a deep-learning model using the 1 Billion Dataset. Sufficient work has been undertaken in this dissertation to prove the potential of this work.

One of the more surprising results observed was that the performance increase flattened off once the size of the training dataset increased beyond 10k. It would be interesting to investigate whether this pattern is observed with other citation parsing tools trained on the 1 Billion Dataset. Further, it would be interesting to investigate how the performance of a model changes when a hand-labelled training dataset is increased in size. This could indicate whether it is the model or the training data that causes the performance to flatten off when the training data goes above 10k.

Future work could also investigate the effects on performance of more or less fine-grained training data. Here, experiments could be conducted with citation parsing tools with different levels of granularity for their training data.

A disadvantage of the 1 Billion Dataset as highlighted in the Discussion,

Chapter 6, is the use of the <container-title> tag. By using the citation type lookup index it is possible to map the <container-title> tag to more meaningful labels such as: journal, book, conference-paper etc. This method was used to make the 1 Billion Dataset compatible with GROBID.

Future work could be done to carry out this mapping for the entire dataset and replace the <container-title> tag with appropriately named tags. This would make the dataset easier to use and reduce the labour associated with converting from one training data format to another.

Given that a citation parsing tool will ultimately have to parse real-world citation strings, many of these citation strings will contain errors. Therefore another potential area for future work would be to introduce some noise into the 1 Billion Dataset and examine the effect this has on performance. Noise could include: extra white spaces, OCR errors, extra or missing punctuation. Noisy training data may improve the ability of a citation parsing tool to parse real-world citation strings.

Chapter 10

Appendices

10.1 Project Repository

All code and documentation associated with this project can be found at the following repository: https://github.com/BeelGroup/1-Billion-Citation-Dataset

10.2 Citation Types

Table 10.2 gives a complete breakdown of all citation types contained within the 1 Billion Dataset.

Citation Type	Number of Citations	Percentage
article-journal	752,005,608	75.9%
chapter	122,562,727	12.4%
paper-conference	55,706,868	5.6%
dataset	17,003,027	1.7%
reference-entry	8,371,603	0.8%
book	7,077,100	0.7%
journal-issue	6,274,342	0.6%
report	5,843,498	0.6%
monograph	3,970,516	0.4%
other	3,507,453	0.4%
dissertation	2,597,706	0.3%
standard	2,397,654	0.2%
reference-book	2,063,251	0.2%
posted-content	1,011,099	0.1%
journal	306,900	0.0%
proceedings	196,686	0.0%
peer-review	184,286	0.0%
report-series	126,501	0.0%
book-part	106,274	0.0%
book-section	51,504	0.0%
book-series	20,172	0.0%
journal-volume	12,296	0.0%
proceedings-series	9,366	0.0%
book-set	4,663	0.0%
Total	**991,411,100**	**100%**

Table 10.1: Complete breakdown of Citation Types contained within the 1 Billion dataset

10.3 Indexes

Table 10.2 shows the complete list of all citation types contained within the 1 Billion Dataset along with their respective indexes.

Index	Citation Type
0	article
1	article-magazine
2	article-newspaper
3	article-journal
4	bill
5	book
6	broadcast
7	chapter
8	dataset
9	entry
10	entry-dictionary
11	entry-encyclopedia
12	figure
13	graphic
14	interview
15	legislation
16	legal_case
17	manuscript
18	map
19	motion_picture
20	musical_score
21	pamphlet
22	paper-conference
23	patent
24	post
25	post-weblog
26	personal_communication
27	report
28	review
29	review-book
30	song
31	speech
32	thesis
33	treaty
34	webpage
35	proceedings-article
36	reference-entry
37	journal-issue
38	reference-book
39	dissertation
40	posted-content
41	standard
42	other
43	monograph
44	book-part
45	peer-review
46	journal
47	proceedings
48	book-series
49	report-series
50	book-section
51	book-set
52	journal-volume
53	proceedings-series

Table 10.2: Complete list of citation types contained in the 1 Billion Dataset along with their respective indexes

Bibliography

[1] Dominika Tkaczyk, Andrew Collins, Paraic Sheridan, and Joeran Beel. 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 - JCDL '18*, pages 99–108, Fort Worth, Texas, USA, 2018. ACM Press.

[2] Citation Style Language. `https://citationstyles.org/`.

[3] Eugene Garfield and Robert King Merton. Citation Indexing-Its Theory and Application in Science, Technology, and Humanities. *Technology and Culture*, 21(4):714, October 1980.

[4] Filippo Radicchi, Santo Fortunato, Benjamin Markines, and Alessandro Vespignani. Diffusion of scientific credits and the ranking of scientists. *Phys. Rev. E*, 80(5):056103, November 2009. arXiv: 0907.1050.

[5] F. Radicchi, S. Fortunato, and C. Castellano. Universality of citation distributions: Toward an objective measure of scientific impact. *Proceedings of the National Academy of Sciences*, 105(45):17268–17272, November 2008.

[6] Linda Butler. Explaining Australia's increased share of ISI publications—the effects of a funding formula based on publication counts. *Research Policy*, 32(1):143–155, January 2003.

[7] Alexander Michael Petersen, Santo Fortunato, Raj K. Pan, Kimmo Kaski, Orion Penner, Armando Rungi, Massimo Riccaboni, H. Eugene Stanley, and Fabio Pammolli. Reputation and impact in academic careers. *PNAS*, 111(43):15316–15321, October 2014.

[8] Scopus - Document search. `https://www.scopus.com`.

[9] Web of Science. `www.webofknowledge.com/`.

[10] Google Scholar. `https://scholar.google.com/`.

[11] J. E. Hirsch. Does the h index have predictive power? *PNAS*, 104(49):19193–19198, December 2007.

[12] Leo Egghe. Theory and practise of the g-index. *Scientometrics*, 69(1):131–152, October 2006.

[13] Animesh Prasad, Manpreet Kaur, and Min-Yen Kan. Neural ParsCit: a deep learning-based reference string parser. *Int J Digit Libr*, 19(4):323–337, November 2018.

[14] David A. King. The scientific impact of nations. *Nature*, 430(6997):311, July 2004.

[15] RefSeek - Academic Search Engine. `https://www.refseek.com/`.

[16] Microsoft Academic. `https://academic.microsoft.com/`.

[17] J. Beel, A. Aizawa, C. Breitinger, and B. Gipp. Mr. DLib: Recommendations-as-a-Service (RaaS) for Academia. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–2, June 2017.

[18] D. Tkaczyk, P. Szostek, P. J. Dendek, M. Fedoryszak, and L. Bolikowski. CERMINE – Automatic Extraction of Metadata and References from Scientific Literature. In *2014 11th IAPR International Workshop on Document Analysis Systems*, pages 217–221, April 2014.

[19] X. Zhang, J. Zou, D. X. Le, and G. R. Thoma. A Structural SVM Approach for Reference Parsing. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 479–484, December 2010.

[20] Hui Han, C. L. Giles, E. Manavoglu, Hongyuan Zha, Zhenyue Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines. In *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, pages 37–48, May 2003.

[21] Kristie Seymore, Andrew McCallum, and Ronald Rosenfeld. Learning Hidden Markov Model Structure for Information Extraction. page 6.

[22] Kaustubh Deshmukh Borkar, Vinayak and Sunita Sarawagi. Automatic segmentation of text into structured records. *ACM SIGMOD Record, 2001*, pages 175–186.

[23] A. Takasu. Bibliographic attribute extraction from erroneous references based on a statistical model. In *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, pages 49–60, May 2003.

[24] Erik Hetzner. A simple method for citation metadata extraction using hidden markov models. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries - JCDL '08*, page 280, Pittsburgh PA, PA, USA, 2008. ACM Press.

[25] Isaac G Councill, C Lee Giles, and Min-Yen Kan. ParsCit: An open-source CRF reference string parsing package. page 7.

[26] Patrice Lopez. GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications. In Maristella Agosti, José Borbinha, Sarantos Kapidakis, Christos Papatheodorou, and Giannis Tsakonas, editors, *Research and Advanced Technology for Digital Libraries*, volume 5714, pages 473–474. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[27] Matteo Romanello, Federico Boschetti, and Gregory Crane. Citations in the digital library of classics: extracting canonical references by using conditional random fields. page 8.

[28] D. Matsuoka, M. Ohta, A. Takasu, and J. Adachi. Examination of effective features for CRF-based bibliography extraction from reference strings. In *2016 Eleventh International Conference on Digital Information Management (ICDIM)*, pages 243–248, September 2016.

[29] Nguyen Viet Cuong, Muthu Kumar Chandrasekaran, Min-Yen Kan, and Wee Sun Lee. Scholarly Document Information Extraction using Extensible Features for Efficient Higher Order Semi-CRFs. In *Proceedings of the 15th ACM/IEEE-CE on Joint Conference on Digital Libraries - JCDL '15*, pages 61–64, Knoxville, Tennessee, USA, 2015. ACM Press.

[30] M. Ohta, D. Arauchi, A. Takasu, and J. Adachi. Empirical Evaluation of CRF-Based Bibliography Extraction from Reference Strings. In *2014 11th IAPR International Workshop on Document Analysis Systems*, pages 287–292, April 2014.

[31] Qing Zhang, Yong-Gang Cao, and Hong Yu. Parsing citations in biomedical articles using conditional random fields. *Computers in Biology and Medicine*, 41(4):190–194, April 2011.

[32] Danny Rodrigues Alves, Giovanni Colavizza, and Frédéric Kaplan. Deep Reference Mining From Scholarly Literature in the Arts and Humanities. *Front. Res. Metr. Anal.*, 3, 2018.

[33] Dominika Tkaczyk, Paraic Sheridan, and Joeran Beel. ParsRec: Meta-Learning Recommendations for Bibliographic Reference Parsing. *arXiv:1808.09036 [cs]*, August 2018. arXiv: 1808.09036.

[34] Citation Style Language. https://citationstyles.org/.

[35] C. Chen, K. Yang, C. Chen, and J. Ho. BibPro: A Citation Parser Based on Sequence Alignment. *IEEE Transactions on Knowledge and Data Engineering*, 24(2):236–250, February 2012.

[36] Kazem Taghva and Eric Stofsky. OCRSpell: an interactive spelling correction system for OCR errors in text. *IJDAR*, 3(3):125–137, March 2001.

[37] Sam Anzaroot and Andrew McCallum. A New Dataset for Fine-Grained Citation Field Extraction. page 8.

[38] Mateusz Fedoryszak, Dominika Tkaczyk, and Łukasz Bolikowski. Large scale citation matching using Apache Hadoop. *arXiv:1303.6906 [cs]*, March 2013.

[39] Alexey Ya. Chervonenkis. Early History of Support Vector Machines. In Bernhard Schölkopf, Zhiyuan Luo, and Vladimir Vovk, editors, *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 13–20. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[40] R. L. Stratonovich. Conditional Markov Processes. In P. I. Kuznetsov, R. L. Stratonovich, and V. I. Tikhonov, editors, *Non-Linear Transformations of Stochastic Processes*, pages 427–453. Pergamon, January 1965.

[41] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Departmental Papers (CIS)*, June 2001.

[42] Warren S. McCulloch and Walter Pitts. A logical Calculus of the Ideas Immanent in Nervous Activity. *The bulletin of mathematical biophysics*, 1943.

[43] Cite this for me. `http://www.citethisforme.com`.

[44] Zotero Style Repository. `https://www.zotero.org/styles`.

[45] Biblio::Citation::Parser - citation parsing framework. `https://metacpan.org/`.

[46] Science Parse parses scientific papers (in PDF form) and returns them in structured form.:. `https://github.com/allenai/science-parse`, June 2019.

[47] Tom M. Mitchell. *Machine learning and data mining*, volume 42. Communications of the ACM, 1999.

[48] Micro Average vs Macro average Performance in a Multiclass classification setting. `https://datascience.stackexchange.com/questions/15989/`.

[49] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. CiteSeer: an automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries - DL '98*, pages 89–98, Pittsburgh, Pennsylvania, United States, 1998. ACM Press.

[50] Elias Kunnas. PDF Structure and Syntactic Analysis for Metadata Extraction and Tagging: https://code.google.com/p/pdfssa4met/ - eliask/pdfssa4met, September 2018. original-date: 2013-03-06T05:29:18Z.

[51] Dustin Heckmann, Anette Frank, Matthias Arnold, Peter Gietz, and Christian Roth. Citation segmentation from sparse & noisy data: A joint inference approach with Markov logic networks. *Lit Linguist Computing*, 31(2):333–356, June 2016.

[52] Eli Cortez, Altigran S. da Silva, Marcos André Gonçalves, Filipe Mesquita, and Edleno S. de Moura. A flexible approach for extracting metadata from bibliographic citations. *Journal of the American Society for Information Science & Technology*, 60(6):1144–1158, June 2009.

[53] Alexandru Constantin, Steve Pettifer, and Andrei Voronkov. PDFX: fully-automated PDF-to-XML conversion of scientific literature. In *Proceedings of the 2013 ACM symposium on Document engineering - DocEng '13*, page 177, Florence, Italy, 2013. ACM Press.

[54] Matthew Richardson and Pedro Domingos. Markov logic networks. *Mach Learn*, 62(1):107–136, February 2006.

[55] Min-Yuh Day, Richard Tzong-Han Tsai, Cheng-Lung Sung, Chiu-Chen Hsieh, Cheng-Wei Lee, Shih-Hung Wu, Kun-Pin Wu, Chorng-Shyong Ong, and Wen-Lian Hsu. Reference metadata extraction using a hierarchical knowledge representation framework. *Decision Support Systems*, 43(1):152–167, February 2007.

[56] Yu-Lun Hsieh, Shih-Hung Liu, Ting-Hao Yang, Yu-Hsuan Chen, Yung-Chun Chang, Gladys Hsieh, Cheng-Wei Shih, Chun-Hung Lu, and Wen-Lian Hsu. A Frame-Based Approach for Reference Metadata Extraction. In Shin-Ming Cheng and Min-Yuh Day, editors, *Technologies and Applications of Artificial Intelligence*, Lecture Notes in Computer Science, pages 154–163. Springer International Publishing, 2014.

[57] Mike Jewell. ParaTools Reference Parsing Toolkit-Version 1.0 Released. *D-lib Magazine*, 9(2), 2003.

[58] Takashi Okada, Atsuhiro Takasu, and Jun Adachi. Bibliographic Component Extraction Using Support Vector Machines and Hidden Markov Models. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Rachel Heery, and Liz Lyon, editors, *Research and Advanced Technology for Digital Libraries*, volume 3232, pages 501–512. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[59] MEDLINE/PubMed Data Documentation. `https://www.nlm.nih.gov/databases/download/pubmed_medline_documentation.html`.

[60] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *Information Processing & Management*, 42(4):963–979, July 2006.

[61] Ping Yin, Ming Zhang, ZhiHong Deng, and DongQing Yang. Metadata Extraction from Bibliographies Using Bigram HMM. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Zhaoneng Chen, Hsinchun Chen, Qihao Miao, Yuxi Fu, Edward Fox, and Ee-peng Lim, editors, *Digital Libraries: International Collaboration and Cross-Fertilization*, volume 3334, pages 310–319. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[62] Bolanle Ojokoh, Ming Zhang, and Jian Tang. A trigram hidden Markov model for metadata extraction from heterogeneous references. *Information Sciences*, 181(9):1538–1551, May 2011.

[63] Angelo Di Iorio, Christoph Lange, Anastasia Dimou, and Sahar Vahdati. Semantic Publishing Challenge - Assessing the Quality of Scientific Output by Information Extraction and Interlinking. *arXiv:1508.06206 [cs]*, August 2015.

[64] CiteSeerX. `https://citeseerx.ist.psu.edu`.

[65] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv:1508.01991 [cs]*, August 2015.

[66] Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF. *arXiv:1603.01354 [cs, stat]*, March 2016.

[67] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural Architectures for Named Entity Recognition. *arXiv:1603.01360 [cs]*, March 2016.

[68] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[69] Young-Min Kim, Patrice Bellot, Jade Tavernier, Elodie Faath, and Marin Dacos. Evaluation of BILBO reference parsing in digital humanities via a comparison of different tools. In *Proceedings of the 2012 ACM symposium on Document engineering - DocEng '12*, page 209, Paris, France, 2012. ACM Press.

[70] Andrew McCallum Data. `https://people.cs.umass.edu/~mccallum/data.html`.

[71] Tudor Groza, AAstrand Grimnes, and Siegfried Handschuh. Reference Information Extraction and Processing Using Random Conditional Fields. *Information Technology and Libraries, Vol 31, Iss 2, Pp 6-20 (2012)*, (2):6, 2012.

[72] GROTOAP-citations - RepOD. `https://repod.pon.edu.pl/dataset/grotoap-citations`.

[73] Sylvester Keil. Anystyle-Parser. `https://github.com/inukshuk/anystyle`, June 2019. original-date: 2011-08-31T16:24:21Z.

[74] Niall Martin Ryan. Citation Data-set for Machine Learning Citation Styles and Entity Extraction from Citation Strings. *arXiv:1805.04798 [cs]*, May 2018.

[75] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[76] Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. page 10.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. pages 770–778, 2016.

[78] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. *arXiv:1412.4729 [cs]*, December 2014. arXiv: 1412.4729.

[79] Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating Text with Recurrent Neural Networks. page 8.

[80] Y. LeCun, Fu Jie Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages 97–104, Washington, DC, USA, 2004. IEEE.

[81] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, April 2007.

[82] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. page 60.

[83] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *Int J Comput Vis*, 77(1-3):157–173, May 2008.

[84] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. page 8.

[85] Mendeley - Reference Management Software & Researcher Network. `https://www.mendeley.com/?interaction_required=true`.

[86] EndNote | Clarivate Analytics. `https://endnote.com/`.

[87] Primer — An Introduction to CSL — Citation Style Language 1.0.1-dev documentation. `http://docs.citationstyles.org/en/1.0.1/primer.html#understanding-csl-styles`.

[88] A JavaScript implementation of the Citation Style Language (CSL) https://citeproc-js.readthedocs.io: Juris-M/citeproc-js, October 2018.

[89] Official repository for Citation Style Language (CSL) citation styles.:. `https://github.com/citation-style-language/styles`, July 2019.

[90] ChrissieCW. Crossref. `https://www.crossref.org/about/`.

[91] Patrice Lopez. A machine learning software for extracting information from scholarly documents:. `https://github.com/kermitt2/grobid`, July 2019.

[92] Bibliographical references - GROBID Documentation. `https://grobid.readthedocs.io/en/latest/training/Bibliographical-references/`.