A High-Quality Gold Standard for Citation-based Tasks

Michael Färber[1,2], Alexander Thiemann[1], Adam Jatowt[2]
[1] University of Freiburg, [2] Kyoto University
michael.faerber@cs.uni-freiburg.de, mail@athiemann.net, adam@dl.kuis.kyoto-u.ac.jp

Abstract

Analyzing and recommending citations within their specific citation contexts has recently received much attention due to the growing number of available publications. Although data sets such as CiteSeerX have been created for evaluating approaches for such tasks, those data sets exhibit striking defects. This is understandable when one considers that both information extraction and entity linking, as well as entity resolution, need to be performed. In this paper, we propose a new evaluation data set for citation-dependent tasks based on arXiv.org publications. Our data set is characterized by the fact that it exhibits almost zero noise in its extracted content and that all citations are linked to their correct publications. Besides the pure content, available on a sentence-by-sentence basis, cited publications are annotated directly in the text via global identifiers. As far as possible, referenced publications are further linked to the DBLP Computer Science Bibliography. Our data set consists of over 15 million sentences and is freely available for research purposes. It can be used for training and testing citation-based tasks, such as recommending citations, determining the functions or importance of citations, and summarizing documents based on their citations.

Keywords: Citations, References, Scholarly Data, Citation Recommendation, arXiv.org, Digital Libraries

1. Introduction

Many tasks concerning digital libraries deal with citations mentioned in scientific texts. These include citation recommendation, which deals with recommending relevant citations for a given citation context. For instance, for the sentence "Models of linear logic have provided a fresh point of view and new intuitions that were applied to traditional fields of study, such as game semantics [?]," the aim would be to recommend appropriate citations such as (Abramsky and McCusker, 1995). Citation recommendation has turned out to be a task attracting increased interest and with significant impact due to the rapidly growing numbers of scientific publications released each year.

Approaches for citation recommendation are mostly evaluated by removing all citations in the considered publications and by letting the tested approach "re-predict" the publications which were cited. In order to allow large-scale experiments, some evaluation data sets have been created, such as CiteSeerX (Caragea et al., 2014). These data sets are sizeable in terms of the number of publications and citation contexts.[1] However, all of them have considerable drawbacks, making it difficult to use those data sets as realistic evaluation data sets – as partially pointed out by (Roy et al., 2016). Two of those drawbacks are (1) *the citation contexts are very noisy* and (2) *there is no interlinking or annotation of citations in the text with a noise-free structured representation of the cited publications* (especially across documents). To the best of our knowledge, CiteSeerX is the only data set which provides not only information about references between papers, but also extracted citation contexts for each citation, thereby solving problem (2) to some extent. However, the citation contexts are very noisy (see Sec. 2.), thereby it suffers from the drawback (1). Furthermore, the CiteSeerX data set contains not only publications, but any manuscripts, since it is built based on crawling web pages. A cleaner data set was created by Carageas et al. (Caragea et al., 2014) based on CiteSeerX, but this data set still does not reach the desired quality for a real-world evaluation of citation recommendation and other tasks.

In this paper, we propose a newly-created gold standard data set for citation-based tasks. This gold standard is based on all computer science papers in arXiv.org and is of very high quality: (1) the extracted sentences are almost always clean and complete, and (2) 100% of the citations in the text are linked to their correct publications. This is due to the fact that for each citation in TEX we know which cited publication is referenced and that we will not miss any citations due to explicitly given `cite` commands. Besides the fact that arXiv.org is a valuable source, arXiv.org is also being used with increasing frequency,[2] making our data set creation approach even more promising in the future.

This paper details how we created the data set and how it can be used. The data set files and associated key figures can be obtained for research purposes at `http://www.citation-recommendation.org/publications/`.[3]

2. Existing Data Sets

CiteSeerX can be regarded as the most frequently used evaluation data set for citation-based tasks. The first version of CiteSeerX was published in 1998 under the name CiteSeer (Giles et al., 1998) and presented a sample of 5,000 documents. For our investigation, we use the snapshot of the entire CiteSeerX dataset as of October 2013,

[1] For instance, the CiteSeerX version by (Huang et al., 2015) consists of over 1M papers and 10.8M citation contexts. As of May 2013, CiteSeerX had up to 52M citations from up to 2M documents (Caragea et al., 2014).

[2] See `https://arxiv.org/year/cs/17`, `https://arxiv.org/year/cs/16`, and so forth.

[3] Note that most articles in arXiv are submitted with the default arXiv license which grants arXiv a perpetual, non-exclusive license. For our data set, we follow this licensing and refer to `https://arxiv.org/help/bulk_data`

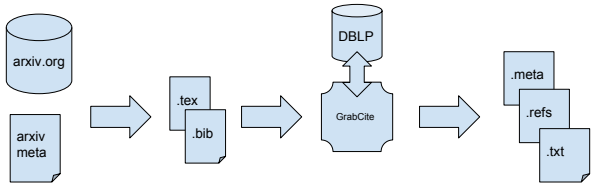

Figure 1: The pipeline used for creating our data set.

published in 2015 by (Huang et al., 2015). This data set consists of 1,017,457 papers, together with 10,760,318 automatically extracted citation contexts. Based on this data set, we can outline the most significant drawbacks of CiteSeerX as follows (cf. also (Roy et al., 2016)):

1. The provided meta-information about cited publications is often not accurate. In particular, the information about the title, the authors, and the venue of cited publications are sometimes incorrectly segmented. Furthermore, a publication's title can be mixed with information about the venue or with the header of the first content paragraph.
2. The citation contexts can contain noise from non-ASCII characters, formulas, section titles, missed references and/or other "unrelated" references, and do not begin with a complete word; instead, a cut-off at a fixed character length position is used.
3. The actual citation in a context is marked with delimiters ("=–=" and "–=–"), but sometimes characters or symbols from preceding words are included.
4. It also seems to be rather difficult to recover the original text of a given paper – meaning that one is essentially limited by the the length of citation context.

Beside CiteSeer and CiteSeerX, there are other collections of scientific publications. Among them are the ACL Anthology corpus (Bird et al., 2008) and Scholarly Dataset 2 (Sugiyama and Kan, 2015). Note that these data sets only contain the publications themselves, typically in PDF format. Therefore, using such data sets for citation-based task evaluation is troublesome, since one must preprocess the data (i.e., (1) extract the content without introducing too much noise, (2) build global identifiers for cited papers, and (3) annotate citations with those identifiers.) Last but not least, data sets for evaluating paper recommendation tasks, such as CiteULike,[4] only provide information on a document level, but no citation contexts.

3. Data Set Creation

The workflow for creating the proposed corpus of arXiv.org publications annotated with citations is presented in Fig. 1. The basic procedure is as follows: We first downloaded all arXiv source files, which are provided by arXiv via Amazon S3.[5] The provided data consists of multiple tar file bundles. Each tar file contains the files of the individual publications. A paper is either a single LaTeX-file, or a compressed folder containing a LaTeX file (at least one), optionally a bibtex file and other resources. We then use the metadata API of arXiv.org[6] to determine the domain of the paper (e.g., "CS" for computer science). In this data set, we only include papers of the computer science domain in order to be able to retrieve meta-information about those papers from DBLP.

The next step consists of processing each individual paper stored either in a single file or in a compressed folder. In the single file case, we directly parse the LaTeX file into a simple abstract syntax tree (AST). Otherwise, we uncompress the folder, identify all bibtex and LaTeX files, and parse them as described in Section 3.1. Then, we traverse both ASTs (the LaTeX AST and the bibtex AST if available) to extract title, text body, references and citations from the paper and represent it in a structured way.

Having obtained the references of each paper, we attempt to generate a globally unique (descriptive) ID for all references and all papers. In the optimal case, this is the DBLP URL of the paper/reference. This step is outlined in Sec. 3.2.

After having obtained all identifiers for all citations as the "offline step," we replace all citation markers (e.g., "`\cite{FooBar}`") with the global publications' identifiers and split the body text of all publications into single sentences (See Sec. 3.3.). For each publication, its sentences (annotated by identifiers) are written to a plain text file. In total, three files were created for each considered publication: A plain text file containing all sentences with global citation identifiers (each sentence on one line), a file with meta-information about this paper, and a file with mappings between the global citation identifiers (used in citations) and the titles of the cited publications (as written in the citing document). The full data format is described in Sec. 5.

In the following sections, we provide more details concerning key steps of our pipeline. The full pipeline is implemented in Haskell in our tool GrabCite, which is freely available on GitHub.[7] Note that in the following, all code snippets are simplified for clarity and brevity.

3.1. Parsing TeX

Working directly on TeX files instead of PDF enables us (1) to know with the utmost certainty the corresponding reference for each citation, and (2) to not miss any citations.

However, using TeX is non-trivial: LaTeX is a very complicated format to parse (Knuth, 1984). Among other things, this is due to the fact that LaTeX is fully customizable and programmable (cf. Turing completeness). Thus, few LaTeX parsers are sufficiently accurate and fast while producing an accessible AST for further processing. For example, the existing Haskell libraries like the popular pandoc (Krijnen et al., 2014) or HaTeX[8] quickly failed on most arXiv documents in our experiments. Most other tools for LaTeX parsing invoke the LaTeX-Engine and work on DVI outputs.[9] Math constructs such as equations are hard to write linearly in text and, moreover, they are not needed for tasks such

[4] See `http://citeulike.org/`.

[5] See `https://arxiv.org/help/bulk_data`.

[6] See `https://arxiv.org/help/oa/arXiv_meta_format.html`.

[7] See `https://github.com/agrafix/grabcite`.

[8] See `https://github.com/Daniel-Diaz/HaTeX`.

[9] See, for instance, TeX4ht, `http://www.tug.org/tex4ht/`.

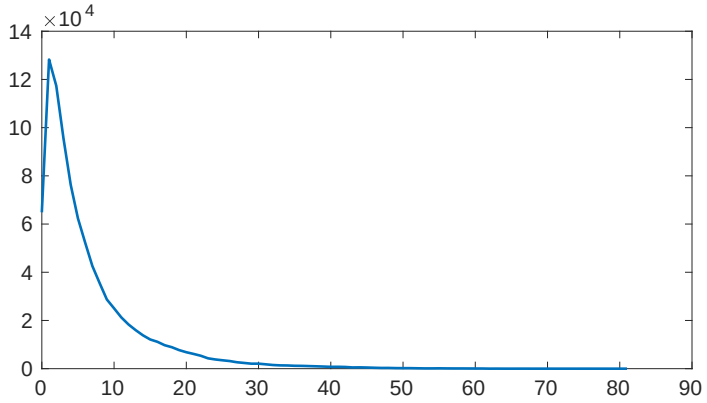

Figure 2: Distribution of time differences between the citing papers and the cited papers in years in our data set.

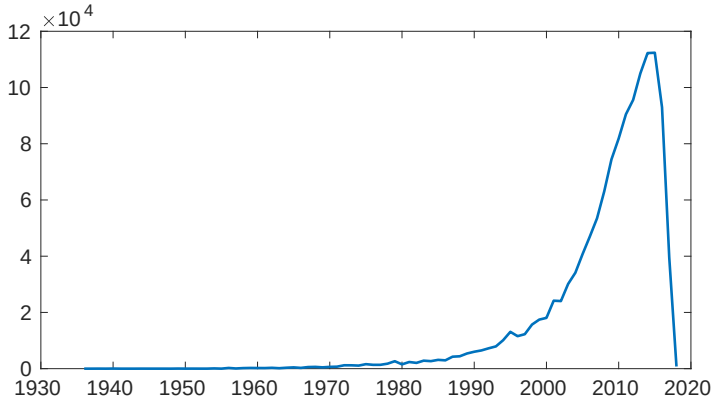

Figure 3: Distribution of the publication dates of all references in our data set.

as citation recommendation. Indeed, mathematical expressions rather disturb the generic learning of machine learning models. Therefore, we ignore math constructs and similar content for our needs.

Our TEX parser is implemented using parser combinators from the Haskell megaparsec library (Karpov, 2015). With the parsed AST at hand, we extract title, text bodies, and references and represent those in a structured way.

All references are written to a map, with the citation key as key and the title and authors as value. The title of the paper is also trivially extracted from the title LATEX command.

Evaluations showed that our extraction method currently fails on 24,762 of all 115,040 TEX input files (21.5%). Multiple factors account for these failures:

- The corresponding TEX input file only includes a PDF; hence, there is no raw content.
- Our heuristic algorithm picked the wrong TEX file from a zip archive.
- Our TEX parser fails due to unimplemented features in our parser.
- The TEX is invalid.

Note, however, that failures result in empty output files. Hence, the high quality of our gold standard is maintained.

3.2. ID Generation

In the ID Generation step, we want to replace local citation markers (e.g., `\cite{FooBar}`) and their associated reference with global identifiers (e.g., `DBLP:http://dblp.org/rec/journals/mscs/Berline06`). We also want to annotate the source papers with DBLP URLs in order to have meta-information about them, as well as graph-based statistics. To obtain the DBLP URL for a given title or reference, we generate a search string for DBLP by tokenizing the input and by using the first n words longer than two characters, thereby increasing n until we have 40 or more characters. We also look for year numbers and include them in the query. In our experiments, this returned the most accurate search results, as noisy words were removed and the query strings were not too long. If the search query returns a result, we use that. Otherwise, we query our own full text search index[10] of the DBLP XML dump[11] (Ley, 2009) which indexes titles and authors. We found that there are some cases where the DBLP API search does not return any results, but our custom full text search does. This full text search is based on PostgreSQL's built in `tsquery` and `similarity` functionality. If this still does not return any meaningful results, we generate as a fallback option our own global ID. First, we look for identifiers like the DOI or the arXiv.org ID to use, and if those are not detectable, we generate an identifier by extracting all words longer than two characters, sorting them, and taking the first 5 in concatenation. In Section 4., we outline the distribution of used references.

[10] See `https://github.com/agrafix/papergrep`.
[11] As of February 15, 2018.

3.3. Sentence Tokenizing

The final step is breaking the input into sentences. We implemented a custom sentence-splitting step due to the fact that many existing sentence tokenizers like `sent_tokenize` from NLTK (Loper and Bird, 2002) became confused by our global citation identifiers. Our sentence splitting uses heuristics to identify abbreviations, numberings, and ellipses, and can correctly handle our global citation markers. It also uses some basic metrics such as word count, character count and punctuation-to-character ratio to detect invalid sentences and to remove them.

4. Data Set Key Figures

Our data set, based on all arXiv.org publications in the computer science domain published until December 31, 2017, contains 90,278 papers. 62,337 (69%) of the papers could be found on DBLP and have been assigned the corresponding DBLP URL in the meta file. We extracted 15,530,204 sentences, resulting in 172 sentences per paper on average. 1,822,836 (11.7%) sentences contain at least one reference. All papers reference 277,227 unique papers using 2,448,826 citation markers in total (i.e., on average 27.1 citation markers per citing paper). Of these references, 962,084 could be found on DBLP and we could assign them a DBLP URL. Furthermore, the 90,278 citing papers cited 18,045 papers which are already in our arXiv data set (i.e., within-arXiv citations; in total, 153,555 single citations), while 259,182 (unique) cited papers are outside of our arXiv data set. For this calculation, we only considered papers with DBLP URLs, so that the value is likely to be under-approximated.

The temporal difference between a citing paper and a cited paper (see Fig. 2) is on average 6.7 years. For over half of all citations (53.1%), the cited paper is at most five years older than the citing paper. The largest gap is 81 years, with the oldest paper referenced having been written in 1936.

In Fig. 3, we show the distribution of the publication dates of all cited papers, with the oldest papers (from 1936) on the left and the most recent papers (from 2018) on the right. In total, 269,194 different authors and 1,489 different publication venues are referenced, with the most popular venues being *Computing Research Repository (CoRR)*[12] (citation count: 67,291) and *IEEE Trans. Information Theory*[13](citation count: 41,436). All mentioned key figures are available online.[14]

5. Data Set Format

The data set is provided as a compressed folder. The folder contains three documents per processed paper: a `.txt`, a `.meta`, and a `.refs` file. The name of each file corresponds to the paper's arXiv.org identifier.[15]

Listing 1: Example excerpt of an output .txt file.

```
In order to parallelize SGD, the standard
    approach is to employ minibatch
    training, which samples multiple
    examples uniformly at each step.
============
The uniformly sampled minibatch stochastic
    gradient is an unbiased estimation of
    the true gradient <DBLP:http://dblp.org
    /rec/conf/icml/Zhang04> <DBLP:http://
    dblp.org/rec/conf/aistats/RakhlinSS13>
    <DBLP:http://dblp.org/rec/conf/icml/
    Shamir013> <DBLP:http://dblp.org/rec/
    journals/jmlr/DuchiS09>, but the
    resulting estimator may have relatively
     high variance.
============
Throughout this paper, we will denote <
    formula> as <formula> for simplicity.
============
[..]
```

Listing 2: Example excerpt of an output .refs file.

```
DBLP:http://dblp.org/rec/conf/icml/ZhaoZ15;
    Peilin Zhao and Tong Zhang. Stochastic
    optimization with importance sampling.
    , abs/1401.2753, 2014.;
DBLP:http://dblp.org/rec/conf/icml/Zhang04;
    Tong Zhang. Solving large scale linear
    prediction problems using stochastic
    gradient descent algorithms. In ICML,
    2004.;
[..]
```

The `.txt` file contains all sentences extracted from the original paper, with local citation markers replaced with our global citation markers. There is one sentence per line, followed by a line containing a separator as shown in Listing 1. This allows the files to be easily skimmed by a human reader while also remaining optimal for machines parsing. Formulas and variables entered in math mode are represented by a `<formula>` token. Figures, tables and other listings, as well as the corresponding captions of the original input, are ignored and cannot be found in the output file.

The `.refs` file (see an example in Listing 2) contains a delimiter-separated dictionary mapping all global citation markers to their original reference descriptions. This al-

Listing 3: Example excerpt of an output .meta file.

```
{ "url":"http://dblp.org/rec/journals/corr/
    ZhaoZ14b"
, "authors":["Peilin Zhao","Tong Zhang"]
, "title":"Accelerating Minibatch
    Stochastic Gradient Descent using
    Stratified Sampling."
}
```

[12] See `https://arxiv.org/corr/home/`.

[13] See `http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?reload=true&punumber=18`.

[14] See `http://www.citation-recommendation.org/publications/`.

[15] See `https://arxiv.org/help/arxiv_identifier`

lows users of the data set to search for the paper in other sources if a DBLP identifier could not be determined by our processing pipeline.

The `.meta` file contains a JSON Document which is generated from data extracted from the paper merged with metadata returned from a search for the document in DBLP. It contains basic metadata such as the title of the paper, the authors of the paper and the DBLP URL of the paper. Note that the DBLP URL is very useful, as it allows users to download more context and metadata corresponding to the paper. For example, we can obtain BibTeX entries, RDF triples and other XML data for each paper using the provided URL. An example of a `.meta` file can be seen in Listing 3.

6. Conclusions

Approaches for citation-based tasks, especially those using machine learning, require clean, high-quality data sets. In this paper, we proposed a new high-quality data set for this purpose: The data set contains 15.5 million sentences of arXiv.org publications in the computer science domain. In those sentences, the citation markers were replaced by global paper identifiers. All citing and cited papers are linked to DBLP as much as possible. The data set can be used for a variety of citation-based tasks, such as citation recommendation, citation function determination, and citation-based document summarization.

In the future, besides improving our TEX parser, we will explore how to link arXiv.org papers to further established identifiers besides DBLP identifiers in order to incorporate arXiv.org papers from further disciplines into our data set.

Acknowledgments. Michael Färber is an International Research Fellow of the Japan Society for the Promotion of Science (JSPS). The work was partially supported by MIC SCOPE (171507010).

7. Bibliographical References

Abramsky, S. and McCusker, G. (1995). Games and Full Abstraction for the Lazy lambda-Calculus. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science*, pages 234–243.

Bird, S., Dale, R., Dorr, B. J., Gibson, B. R., Joseph, M. T., Kan, M., Lee, D., Powley, B., Radev, D. R., and Tan, Y. F. (2008). The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the International Conference on Language Resources and Evaluation*, LREC 2008.

Caragea, C., Wu, J., Ciobanu, A. M., Williams, K., Ramírez, J. P. F., Chen, H., Wu, Z., and Giles, C. L. (2014). CiteSeer x : A Scholarly Big Dataset. In *Proceedings of the 36th European Conference on IR Research*, ECIR 2014, pages 311–322.

Giles, C. L., Bollacker, K. D., and Lawrence, S. (1998). CiteSeer: An Automatic Citation Indexing System. In *Proceedings of the 3rd ACM International Conference on Digital Libraries*, pages 89–98.

Huang, W., Wu, Z., Chen, L., Mitra, P., and Giles, C. L. (2015). A Neural Probabilistic Model for Context Based Citation Recommendation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2404–2410.

Karpov, M. (2015). Hackage: megaparsec. `https://http://hackage.haskell.org/package/megaparsec`. Accessed: 2017-09-21.

Knuth, D. E. (1984). A Torture Test for TEX. Technical report, Stanford, CA, USA.

Krijnen, J., Swierstra, S. D., and Viera, M. (2014). Expand: Towards an Extensible Pandoc System. In *Practical Aspects of Declarative Languages – 16th International Symposium*, PADL 2014, pages 200–215.

Ley, M. (2009). DBLP - some lessons learned. *PVLDB*, 2(2):1493–1500.

Loper, E. and Bird, S. (2002). NLTK: the natural language toolkit. *CoRR*, cs.CL/0205028.

Roy, D., Ray, K., and Mitra, M. (2016). From a Scholarly Big Dataset to a Test Collection for Bibliographic Citation Recommendation. In *Scholarly Big Data: AI Perspectives, Challenges, and Ideas, Papers from the 2016 AAAI Workshop*.

Sugiyama, K. and Kan, M. (2015). A comprehensive evaluation of scholarly paper recommendation using potential citation papers. *Int. J. on Digital Libraries*, 16(2):91–109.