A Supervised Learning Approach To Entity Matching Between Scholarly Big Datasets

Jian Wu[1], Athar Sefid[2], Allen C. Ge[1], C. Lee Giles[1,2]
[1]IST, Pennsylvania State University, University Park, PA, 16802 USA
[2]CSE, Pennsylvania State University, University Park, PA, 16802 USA
jxw394@ist.psu.edu,azs5955@cse.psu.edu

ABSTRACT

Bibliography metadata in scientific documents are essential in indexing and retrieval of scholarly big data for production search engines and bibliometrics research studies. Crawl-based digital library search engines can harvest millions of documents efficiently but metadata information extracted by automatic extractors are often noisy, incomplete, and/or with parsing errors. These metadata could be cleaned given a reference database. In this work, we develop a supervised machine learning based approach to match entities in a target database to a reference database, which can further be used to clean metadata in the target database. The approach leverages a number of features extracted from headers available from automatic extraction results. By adjusting combinations of hyper-parameters and various sampling strategies, the best results of Support Vector Machines, Logistic Regression, Random Forests, and Naïve Bayes models give comparable results, with F1-measure of about 90%, outperforming information retrieval only based method by about 14%, evaluated with cross validation.

CCS CONCEPTS

• **Information systems → Clustering and classification**; *Document collection models*; *Information extraction*;

ACM Reference Format:
Jian Wu[1], Athar Sefid[2], Allen C. Ge[1], C. Lee Giles[1,2] [1]IST, Pennsylvania State University, University Park, PA, 16802 USA [2]CSE, Pennsylvania State University, University Park, PA, 16802 USA jxw394@ist.psu.edu,azs5955@cse.psu.edu . 2017. A Supervised Learning Approach To Entity Matching Between Scholarly Big Datasets. In *K-CAP 2017: K-CAP 2017: Knowledge Capture Conference, December 4–6, 2017, Austin, TX, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3148011.3154470

1 INTRODUCTION

Digital library search engines (DLSEs) such as Google Scholar, Microsoft Academic, Semantic Scholar, and CiteSeerX collect data by actively crawling PDF documents from the Web. In most cases, a pipeline is built to distinguish scientific documents from other types of documents, and automatically extract textual and non-textual content from crawled files. Metadata are then parsed and indexed before they become searchable from the web UI or APIs. Among these crawl-based DLSEs, Microsoft Academic and CiteSeerX release scholarly databases. These scholarly big data are widely used in a variety of computer and information science research projects, e.g., [10, 11, 18].

One disadvantage of using these data corpora is that the metadata are noisy at various levels due to extraction errors, which are usually caused by (1) imperfection of metadata extractors; (2) information missing in the crawled PDF files, and (3) heterogeneous layout of open access papers. As a result, it is extremely hard to detect these errors in vast amount of dataset, so most research studies just choose to ignore them, thus reducing the robustness of experimental results and conclusions based on uncleaned datasets. Cleaning noisy digital library data is then essential to mitigate this issue.

Incorrect metadata could be corrected by individual users. The drawback is that only a small proportion of highly visible papers are corrected [17]. One automatic technique leverages reference datasets with reliable data, the source of which usually comes from manual input, or it has been visually inspected and verified. For example, the ACM digital library metadata are manually typed by authors. The procedure is first to link entities in the *target* dataset to entities in a *reference* dataset. The erroneous data are then overwritten by the clean data. The target corpus can also be augmented to include more information from the reference dataset.

There has not been any large scale effort on a machine learning (ML) based software framework to clean automatically extracted metadata and link entities to main stream digital library records. In this paper, we develop an ML based approach to match entities in the target dataset to a reference dataset. Although supervised ML requires laborious labeling work, we feel it is necessary at the this stage because interacting with real data is effective and fundamental to understand data itself before going further to build advanced methods to overcome the limitation of supervised learning.

2 RELATED WORK

There has been much work on record linkage [3, 5]. Most work focus on either nominal or numerical entities, and heavily rely on information retrieval (IR) methods. To our knowledge, there has not been application of machine learning (ML) algorithms on bibliography records in digital libraries.

In [1], the authors created a scholarly big dataset by matching CiteSeerX against DBLP datasets. The reference dataset (DBLP) was indexed by Apache Solr and various attributes of entities from the target dataset (CiteSeerX) are experimented to compare matching performance based on manually labeled samples. It was found

K-CAP 2017, December 4–6, 2017, Austin, TX, USA

ACM ISBN 978-1-4503-5553-7/17/12…$15.00
https://doi.org/10.1145/3148011.3154470

that using 3-gram of titles and Jaccard similarity with a threshold of 0.7 achieves the best F1-measure of about 0.77. However, this approach cannot be directly used in cleaning CiteSeerX data due to the relatively low precision. Even in the best scenario, ∼ 23% records may be corrected "by mistake".

Another work dedicated on cleaning the CiteSeer data describes a local and an online matching algorithm [13]. The authors claim they obtain a F1-measure of 0.96 based, significantly better than the result by [1]. When attempted to repeat their experiments, we could not achieve the same performance. With lack of matching details and response from the authors, the results cannot be verified. While hash values are widely used to provide a unique encryption to a document, *simhash* is especially useful because similar documents are close in terms of simhash distances. The *simhash* encodes a long string to a fixed size fingerprint [2, 14]. We use *simhash* to encrypt titles and abstracts then calculate Hamming distances of encrypted strings. Supervised ML has been applied to author name disambiguation, e.g., [6, 12]. The features include a number of similarity profiles such as author similarities, affiliation similarity, co-authors similarity, concept similarity, journal similarity, and paper title similarity. Several supervised ML models are compared and random forest (RF) achieves the best performance. Our work is inspired by this methodology except that we are linking paper entities in *two* databases.

3 METHODS

3.1 Problem Definition and Challenges

Our problem can be formalized in the following way. We denote the target corpus, which contain noisy data, as T, containing n entities $T = \{t_1, t_2, \cdots, t_n\}$, and the reference corpus, which contain reference data, as R, containing m entities $R = \{r_1, r_2, \cdots, r_m\}$. Each entity can be represented by a number of features, i.e.,

$$t_i = (f_1, f_2, \cdots, f_k), r_j = (f_1, f_2, \cdots, f_k)$$

, in which $(f_1, f_2, \cdots, f_k)$ can be extracted from attributes of t_i or r_j. The goal is to find a set M,

$$M = \{(t, r); t = r, t \in T, r \in R\}$$

. To achieve the goal, we find a set of features $\{f_i\}$ that appropriately represent the entity. The task poses challenges in multiple aspects. (1) A primary key is not always available. In particular, the digital object identifier (DOI), does not always exist for papers crawled from the web, a majority of which are manuscripts. In a random sample of 1000 CiteSeerX papers, only 57 contain DOIs. As a result, in most cases, we must rely on non-primary attributes. Empirically, we found that (title, authors, year, venue) can be used as a composite key to unique identify a paper. (2) It is unknown which fields contain noisy and incomplete data in advance. Data that are used for matching may also be noisy. Similarity-based comparisons and data normalization are applied to mitigate this problem. (3) Pairwise comparison is infeasible across all elements between two databases due to the quadratic complexity. Similar to author name disambiguation [12], it is desired to find a way of narrowing down search space to make the algorithm scalable. We narrow down search space by querying documents indexed by a search platform. Typically, querying a target document t results in a list of reference $\{r_i\}, 1 \le i \le k$, which results in k matching pairs $\{(t, r_i)\}$. Features are derived by comparing corresponding fields of t and r_i.

3.2 Feature Extraction

The following features are extracted from a candidate pair (t, r_i).

(1) Levenshtein distance of simhash values of normalized titles. For both target and reference corpora, the titles are normalized in the following way: (1) all letters are converted to lowercase; (2) characters with diacritics are converted to corresponding ASCII letters; (3) multiple spaces are collapsed to a single space; (4) punctuation marks are removed; (5) single letters "s" and "t" are removed, which are mostly result from removing apostrophe from possessives or abbreviations, e.g., `can't`. The normalized title is then encrypted by simhash [2] to a 16 byte string containing alphanumeric characters.

(2) Levenshtein distance of simhash values of abstract. We do not normalize abstracts because they are usually much longer than titles and normalization does not significantly enhance similarity values.

(3) Year similarity represented by the absolute difference. Year information may be missing, in which it is set to −1. In these cases, the year difference is enormously large. For training purposes, we set the difference to be 100 when the it is greater than 100.

(4) First author similarity represented by a three-digit binary (lmf). Each digit represents whether the last name l, the middle initial m, and the first initial f matches or not. If a certain field is missing or it does not match with the corresponding name part of r_i, the binary is set to 0. The decimal values of the binary are used as feature values. Author names are also normalized before comparison. In addition to converting diacritic characters to corresponding alphabets, and converting all letters to lowercase, prefixes such as "Prof.", "Dr.", and their variants are removed. Suffixes such as "II" are also pruned.

Of the first authors: (5) First name similarity. The feature value is 1 if one of the first names is missing. It is 0 if both first names exist but are different. It is 3 if full first names are available and they are equal. It is set to 2 if only the first name initials are available and they are equal. **(6) Middle name similarity**, determined in the similar way as (5). **(7) Last name similarity**, determined in the similar way as (5) except that it never equals to 2.

Of the last authors: (8) first name similarity, determined in the similar way as (5). **(9) Middle name similarity**, determined in the similar way as (5). **(10) Last name similarity**, determined in the similar way as (7).

(11) All authors' last names similarity represented by Jaccard similarity $(L_t \cap L_r)/(L_t \cup L_r)$ in which L_t and L_r stand for the set of last names for a paper in the target corpus and the reference corpus, respectively.

Venue names are resolved by the venue disambiguation package proposed by [9]. However, the fraction of papers with venue information available is too small, so this feature is not adopted.

4 EXPERIMENTS

4.1 Data Preprocessing

As a case study, we adopt the CiteSeerX papers as the target corpus, which contain about metadata of about 10 million scholarly papers. The header information is extracted by SVMHeaderParser [4] [16].

This extractor does not well equally well on papers from all domains [17]. To improve the quality in the first place we re-extract metadata using GROBID, a CRF based tool to extract bibliographic data from scholarly documents [8]. GROBID has shown superior performance over its peers [7]. The outputs are marked up under the TEI schema, including all header fields needed for feature extraction. At this stage, we focus on training the entity linking model, so we randomly sampled 1000 documents from the CiteSeerX database. GROBID successfully extracts metadata and text from 995 documents. PDFMEF [15] was used in extraction tasks.

We use the IEEE Xplore database that we downloaded from its FTP site as the reference corpus, which includes about 2 million academic papers. The IEEE metadata, provided by publishers, are of high accuracy. This corpus is indexed by ElasticSearch, with fields including DOI, title, abstract, venue, year, publisher, volume, keywords, the first, middle, and, last names.

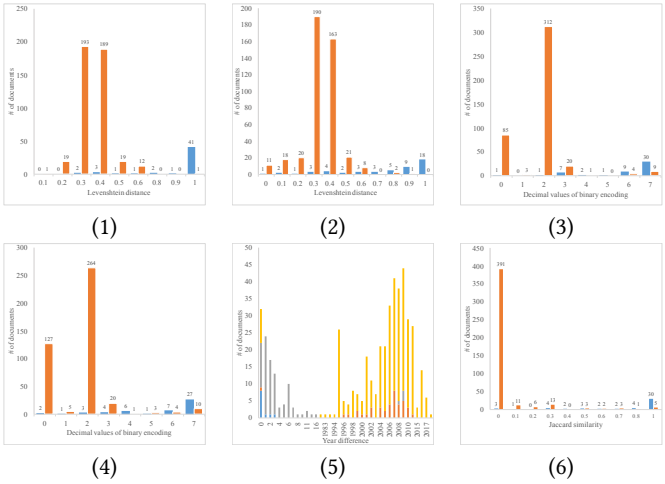


Figure 1: Distribution of documents in feature space. (1) the Levenshtein distance of simhash of normalized titles; (2) the Levenshtein distance of simhash of abstracts; (3) the decimal values of the three-binary encoding of the first author; (4) the decimal values of the three-binary encoding of the last author; (5) the year offset; (6) Jaccard similarity of all authors' last names. Bars are color-coded in the following way: blue – matching pairs with complete data; orange – matching pairs with missing data; gray – non-matching with complete data; yellow – non-matching with missing data.

4.2 Ground Truth Analysis

We employ two graduate students in computer science to independently label matching pairs under the same instructions. First, a list of matching candidates is obtained by querying titles or year+first author's last name of each document. Candidate papers are the top 10 papers returned. The true matching pairs are identified by comparing displayed metadata and the original PDF files. The ones that they disagree upon are judged by a research faculty. The consensus rate, defined as the number of matching pairs they agree on over the total number of unique matching pairs is about 96%. The remaining candidates and the target paper constitutes the negative sample. We finally identified 51 true matching pairs, and 485 non-matching pairs. The relatively low number of matching pairs is likely to be caused by (1) we do not have the complete IEEE collection, estimated to be at least 6 million; (2) the current CiteSeerX collection contains a larger fraction of papers in non-computer science domains. Due to the relatively low number of positive cases, we use 5-fold cross validation (CV) to evaluate the models.

Before building the model, we perform analysis on the labeled dataset. The purpose is to investigate potential correlations between features and classes. For each feature, we plot distribution of document counts of both positive and negative documents in the value space (Figure 1).

Panel (1) clearly exhibits a bimodal distribution, indicating that normalized title is a strong feature to discriminate matching and non-matching pairs. In Panel (2), the positive cases are distributed across a broader range. In Panel (3), while most matching pairs have perfect first author match (value of 7), there are 9 matching pairs that do not match in their first names, and 7 pairs that do not match in their last names (value of 3). Panel (4) exhibits a similar distribution. This indicates that the abstract and author names are useful but not very strong features. Author orders may also not be preserved. Panel (5) shows that the year offsets of matching pairs are up to 3, with the year information missing in most cases. In Panel (6), matching and non-matching papers are well separated, indicating that the Jaccard similarity of author names is another strong feature.

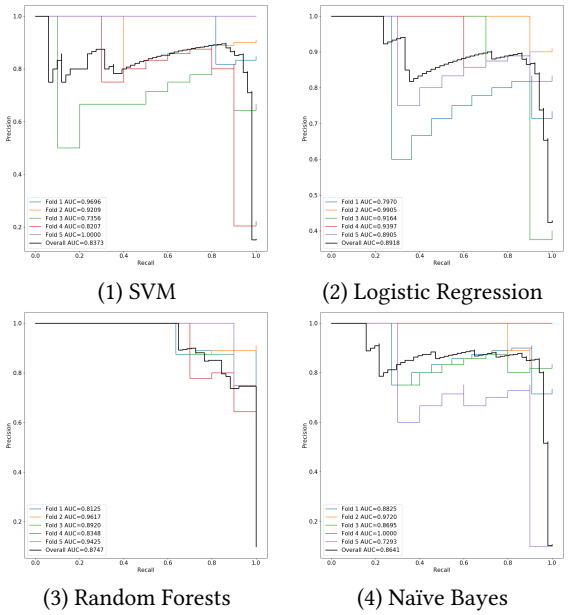
4.3 Data Models

Our task can be formalized as a binary classification problem. We need to build a classifier that performs binary classification given a target paper and a set of matching candidates. Four supervised machine learning models are investigated, support vector machine (SVM), logistic regression (LR), RF, and Naïve Bayes (NB). We explore both over- and under-sampling strategies, attempting to mitigate the bias caused by unbalanced positive and negative samples. The best combination of parameters are determined using grid search. We also compare performances with cases with no sampling. The results are tabulated in Table 1. The models are implemented using scikit-learn (v0.19). Stratified sampling is used when possible. The precision-recall curves are plotted in Figure 2.

Table 1: The 5-fold CV results of models under various sampling strategies.

Model	Sampling Method	Precision	Recall	F1
SVM	No sampling	0.91	0.88	0.90
SVM	Random	0.89	0.94	0.91
LR	No sampling	0.90	0.88	0.89
LR	SMOTE	0.86	0.92	0.89
RF	No sampling	0.91	0.90	0.91
RF	Cluster Centroids	0.90	0.92	0.91
NB	No sampling	0.67	1.00	0.80
NB	Cluster Centroids	0.74	0.92	0.82

The best results by all models, except for NB, perform almost equally well with proper combination of hyper-parameters and sampling strategies. The relatively poor performance of NB could be due to the inter-dependency between features. For example, for



(1) SVM (2) Logistic Regression

(3) Random Forests (4) Naïve Bayes

Figure 2: Precision-recall plots of the four models. Blue, red, green, brown, and purple curves are plots of individual fold. Black curves are the 5-fold composites.

a certain types of papers with uncommon layouts, when titles are not accurately extracted, it is likely that authors are not accurately extracted. Over- or under-sampling strategies may not necessarily outperform cases when no sampling is performed. Compared with matching approaches that are entirely based on IR methods [1], our approach increases the overall F1-measure by about 14%, although CV may raise the performance. A more rigorous comparison will be performed when more labeled data become available.

5 DISCUSSION AND CONCLUSION

While the supervised approach we proposed achieves decent performance, there is still room for improvement. The first is to increase the size of the ground truth dataset. The labeled sample in [1], which is generated by matching CiteSeerX against DBLP (5 million) contain 236 matching pairs. Second, to apply our model to the real data cleaning task, it is desirable to ensure a high precision at the sacrifice of recall because of the high volume of records that needs to be corrected. Therefore, a heuristic method should be applied on top of the supervised approach for the purpose of filtering out extremely unhealthy data. Finally, in case the title extraction has an extremely low quality, it is important to include other metadata fields, such as citations, as features. Our preliminary work has shown promising results.

In summary, we developed a supervised machine learning approach to link entities in a scholarly database with noisy metadata to a reference database. The approach exhibits superior performance over the IR only based method, which can potentially used to improve metadata quality of the target database at scale. One application is to clean the CiteSeerX metadata and investigate its coverage by linking paper entities to other digital libraries in multidisciplinary subject domains.

REFERENCES

[1] Cornelia Caragea, Jian Wu, Alina Ciobanu, Kyle Williams, Juan Fernández-Ramírez, Hung-Hsuan Chen, Zhaohui Wu, and Lee Giles. 2014. *CiteSeerX: A Scholarly Big Dataset.* Springer International Publishing, Cham, 311–322.

[2] Moses Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada.* 380–388.

[3] Peter Christen. 2012. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection.* Springer Science & Business Media.

[4] Hui Han, C. Lee Giles, Eren Manavoglu, Hongyuan Zha, Zhenyue Zhang, and Edward A. Fox. 2003. Automatic document metadata extraction using support vector machines. In *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '03).* 37–48.

[5] Thomas N. Herzog, Fritz J. Scheuren, and William E. Winkler. 2007. *Data Quality and Record Linkage Techniques* (1st ed.). Springer Publishing Company, Incorporated.

[6] Kunho Kim, Madian Khabsa, and C. Lee Giles. 2016. Inventor Name Disambiguation for a Patent Database Using a Random Forest and DBSCAN. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, JCDL 2016, Newark, NJ, USA, June 19 - 23, 2016.* 269–270.

[7] Mario Lipinski, Kevin Yao, Corinna Breitinger, Joeran Beel, and Bela Gipp. 2013. Evaluation of Header Metadata Extraction Approaches and Tools for Scientific PDF Documents. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '13).* ACM, New York, NY, USA, 385–386.

[8] Patrice Lopez. 2009. GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications. In *Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'09).* Springer-Verlag, Berlin, Heidelberg, 473–474.

[9] Denilson Alves Pereira, Eduardo Emanuel Braga da Silva, and Ahmed A. A. Esmin. 2014. Disambiguating publication venue titles using association rules. In *IEEE/ACM Joint Conference on Digital Libraries, JCDL 2014, London, United Kingdom, September 8-12, 2014.* 77–86.

[10] Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Kumar Divvala, and Ali Farhadi. 2016. FigureSeer: Parsing Result-Figures in Research Papers. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII.* 664–680.

[11] Marko Stamenovic, Sam Schick, and Jiebo Luo. 2017. Machine Identification of High Impact Research through Text and Image Analysis. In *Third IEEE International Conference on Multimedia Big Data, BigMM 2017, Laguna Hills, CA, USA, April 19-21, 2017.* 98–104.

[12] Pucktada Treeratpituk and C. Lee Giles. 2009. Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries (JCDL '09).* ACM, New York, NY, USA, 39–48.

[13] Yan Wang, Hao Zhang, Yaxin Li, Deyun Wang, YanLin Ma, Tong Zhou, and Jianguo Lu. 2016. A Data Cleaning Method for CiteSeer Dataset. In *Web Information Systems Engineering - WISE 2016 - 17th International Conference, Shanghai, China, November 8-10, 2016, Proceedings, Part I.* 35–49.

[14] Kyle Williams and C. Lee Giles. 2013. Near Duplicate Detection in an Academic Digital Library. In *Proceedings of the 2013 ACM Symposium on Document Engineering (DocEng '13).* ACM, New York, NY, USA, 91–94.

[15] Jian Wu, Jason Killian, Huaiyu Yang, Kyle Williams, Sagnik Ray Choudhury, Suppawong Tuarob, Cornelia Caragea, and C. Lee Giles. 2015. PDFMEF: A Multi-Entity Knowledge Extraction Framework for Scholarly Documents and Semantic Search. In *Proceedings of the 8th International Conference on Knowledge Capture (K-CAP 2015).* ACM, New York, NY, USA, Article 13, 8 pages.

[16] Jian Wu, Kyle Williams, Hung-Hsuan Chen, Madian Khabsa, Cornelia Caragea, Alexander Ororbia, Douglas Jordan, and C. Lee Giles. 2014. CiteSeerX: AI in a Digital Library Search Engine. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.* 2930–2937.

[17] Jian Wu, Kyle Williams, Madian Khabsa, and C. Lee Giles. 2014. The impact of user corrections on a crawl-based digital library: A CiteSeerX perspective. In *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2014, Miami, Florida, USA, October 22-25, 2014.* 171–176.

[18] Shuai Xiao, Junchi Yan, Changsheng Li, Bo Jin, Xiangfeng Wang, Xiaokang Yang, Stephen M. Chu, and Hongyuan Zha. 2016. On Modeling and Predicting Individual Paper Citation Count over Time. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016.* 2676–2682.