CiteSeerX-2018: A Cleansed Multidisciplinary Scholarly Big Dataset

1st Jian Wu
Computer Science
Old Dominion University
Norfolk, VA, USA
jwu@cs.odu.edu

2nd Bharath Kandimalla
Information Sciences and Technology
Pennsylvania State University
University Park, PA, USA
bkk48@psu.edu

3rd Shaurya Rohatgi
Information Sciences and Technology
Pennsylvania State University
University Park, PA, USA
szr207@psu.edu

4th Athar Sefid
Computer Science and Engineering
Pennsylvania State University
University Park, PA, USA
azs5955@cse.psu.edu

5th Jianyu Mao
Computer Science and Engineering
Pennsylvania State University
University Park, PA, USA
jxm6165@psu.edu

6th C. Lee Giles
Information Sciences and Technology
Pennsylvania State University
University Park, PA, USA
giles@ist.psu.edu

***Abstract*—We report the preliminary work on cleansing and classifying a scholarly big dataset containing 10+ million academic documents released by CiteSeerX. We design novel approaches to match paper entities in CiteSeerX to reference datasets, including DBLP, Web of Science, and Medline, resulting in 4.2M unique matches, whose metadata can be cleansed. We also investigate traditional machine learning and neural network methods to classify abstracts into 6 subject categories. The classification results reveal that the current CiteSeerX dataset is highly multidisciplinary, containing papers well beyond computer and information sciences.**

I. Introduction

Since CiteSeerX was launched in 1998, its data has been used to study properties of citation networks, co-author networks, and for research on keyword extraction, document classification, and recommendation systems. The metadata is obtained from a pipeline built on a set of legacy extraction systems [7], some were trained on data corpus in constrained domains, e.g., computer science [2]. In the past decade, CiteSeerX incorporated academic documents from resources in multiple disciplines, such as physics (from arXiv) and biomedical science (from PubMed) [6]. The heterogeneous nature of document formats across multiple domains makes the legacy extractor producing noisy metadata, containing incomplete fields and incorrect values. A visual inspection on a corpus of user corrected papers indicates that about 30% of titles and 40% of author names have more or less parsing errors [5]. Because samples selected tend to contain wrong metadata, the fractions evaluated on the entire dataset would even be lower. However, other indications imply further data cleansing is necessary.

The cleansing task can be framed to an entity matching problem, in which the noisy dataset (i.e., target dataset) is matched against a clean dataset (i.e., reference dataset). Examples of these datasets include DBLP, Web of Science (WoS), and Medline, whose metadata are originally input by authors or editors. These datasets usually cover a specific type of documents. For example, DBLP covers mostly computer science conference proceedings. Medline covers mostly life science and biomedical science journal papers. WoS covers prestigious journals in various fields but a small fraction of conference proceedings.

The challenge of the entity matching task is that the target dataset is noisy. Previously, an unsupervised method [1] was proposed, which queries n-grams from paper titles against the DBLP metadata indexed by Apache Solr. It found a set of parameters in the best scenario that achieves an F1 of 0.77. The relatively low precision could potentially predict a substantial number of false positives when applying the algorithm on millions of documents. We propose a system combining machine learning and information retrieval methods using header information (i.e., the title, authors, year, abstract) and citations to match against reference datasets. The best F1 is 0.922 using only header information, and 0.992 using both header and citation information. We match the CiteSeerX data against DBLP, WoS, and Medline, and obtained 4.2M unique matched documents, whose metadata can be cleansed.

Text classification is a fundamental task in natural language processing and has been applied in classifying webpages, movie reviews, etc. However, to the best of our knowledge, there has not been work on investigating effective methods for systematically classifying scholarly big data into subject categories (SCs). A major obstacle is the lack of a large scale training corpus. The WoS dataset, containing high quality titles and abstracts of nearly 25 million papers, provides an ideal sample to train robust machine learning and deep learning models. Here, we explore supervised and neural network methods, focusing on relatively less complicated features in order to find a scalable solution. Our results indicate that Logistic Regression and Random Forest achieve a comparable performance to a Multilayer Perceptron model using the same training settings.

II. Cleansing the CiteSeerX Data

The key step to cleansing the CiteSeerX data is to find matching paper entities in a reference dataset. This problem is given one instance t in the target dataset $\mathbb{T}$, find a bibliographic record r from the reference dataset $\mathbb{R}$ for which the similarity between r and t is greater than a threshold. We consider three scenarios: (i) using header information only, (ii) using citations only, (iii) using both header and citations. Three models were built corresponding to each scenario with features and evaluations outlined below. The details are elaborated in [4].

In Scenario (i), the similarity is calculated based on each field in the header. A pairwise comparison between $\mathbb{T}$ and $\mathbb{R}$ requires $|\mathbb{T}| \times |\mathbb{R}|$ comparisons. To narrow the search space, we first index titles, authors, and years of $\mathbb{R}$ and query this index using the title or the author+year of t. The query of author+year is to supplement the cases in which the titles are very short or wrong. The search results are ranked by the standard BM25 algorithm [3] and the first 20 are selected. In the second step, we classify these 20 (t, r) pairs into true and false matches based on 10 similarity based features: (1) Levenshtein distance of simhashes of normalized titles; (2) Levenshtein distance of simashes of abstracts; (3) Jaccard similarity of tokens in normalized titles; (4) Jaccard similarity of tokens in abstracts; (5) Absolute difference of years; (6) The first author's full name similarity; (7) The last author's full name similarity; (8) The first author's last name similarity; (9) The last author's last name similarity; (10) All author's last name similarity. If the classifier predicts no positive sample, then a real match is not found. If the classifier predicts multiple positive samples, we choose the one with the highest BM25 score. We call this HMM (a header matching model).

In Scenario (ii), the matching is based on citations of a paper. Similar to a HMM, it starts with indexing all citations (rather than just papers) in $\mathbb{R}$, called $\mathbb{I}$. Note that a citation record usually contains a title, authors, and a year, but no abstracts. Given a paper t, which have citations $\{tc_i, i = 1, 2, \cdots\}$, the idea is to query tc_j against $\mathbb{I}$ and find a match using HMM. Assuming a matching citation record rc is found, which is cited by r_1, the next step is to compare t and r_1. Our strategy is to calculate the Jaccard similarity between title tokens of all citations of t and r_1. We call this model CMM (citation matching model).

In Scenario (iii), we first attempt to match header information using HMM. If a match is not found, we evaluate the quality of the title string of t. If the quality is low (unlikely to be a title), we propose to match by citations using CMM. Otherwise, no further matching is performed which is denoted as IMM (integrated matching model).

Using this system, we match the CiteSeerX data against three reference datasets: DBLP containing 4M papers, WoS containing 45M papers, and Medline containing 24M papers. The models are trained on a manually labeled dataset containing 688 positive and 1845 negative matching pairs. The matching results are illustrated in Figure 1. Note that we apply IMM only on WoS data because DBLP and Medline data do not contain citations. The total number of CiteSeerX papers whose metadata can be cleansed is about 4.2M.

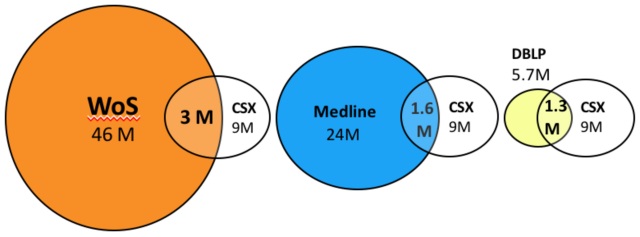

Fig. 1. Matching results between CiteSeerX (CSX) and reference datasets.

III. Classification by Subject Categories

WoS has a Subject Category (SC) scheme that is comprised of 252 SCs in science, social sciences, arts, and humanities. The scheme is created by assigning each journal to one or more SCs. The WoS subject scheme is generally considered the best for bibliometric analysis as its granularity enables users to objectively measure performance against papers similar in scope and citation characteristics. Each published item will inherit all SCs assigned to the parent journal.

In our preliminary study, we focus on classifying documents into 6 SCs: Physics (PHYS), Chemistry (CHEM), Biology (BIO), Materials science (MATSC), Computer Science (CMPSC), and Others using high quality titles and abstracts selected from 25 million papers in WoS. When making the training corpus, we include papers that are assigned with only a single SC. We also collapse subcategories under a broad SC. For example, all papers that are labeled "Computer Science, Artificial Intelligence", "Computer Science, Cybernetics", and " Computer Science and Cybernetics" are grouped under CMPSC. In total, this results in about 1.10M papers classified in PHYS, 1.09M in CHEM, 456k in BIO, 260k in MATSC, and 169k in CMPSC. To balance sample sizes, we downsize each SC corpus to a fixed number $N_{\mathrm{G}} = 150$k. Samples in the "Other" category are randomly selected from documents that are labeled other than the five above. The ground truth dataset is split into a training and a testing corpus, each respectively taking 70% and 30%.

A. Supervised Learning Model

The Bag of Words (BoW) model is one of the most widely used baseline models. It transfers an abstract with variable lengths to a sparse matrix with a fixed number of features, each of which is a unique token where each abstract is tokenized and stemmed. Stopwords in the NLTK stopword list are removed. The TF-IDF is calculated for each token. Each abstract is then represented by a vector, comprised of TF-IDFs of tokens. Four classifiers are trained on this multi-class classification task, namely a support vector machine (SVM), Logistic Regression (LR), Multinomial Naïve Bayes (MNB), and Random Forest (RF). *In this setting, documents are classified into 5 SCs without the Others category.*

In our experiments, we vary the training corpus size N_{G} from 10k to 150k for the 4 classifer models above. The results indicate that LR consistently achieves the best performance given a sample size, followed by SVM. The sample size has little influence on the performance when $N_{\mathrm{G}} > 50$k. The highest micro-F1 is 0.87. MNB achieves 0.84 at best. We also vary the feature vector dimension d_{f} from 1k to 50k for LR. Figure 2 shows that the performance slowly improves when $d_{\mathrm{f}} > 10$k at all sample sizes. The best micro-F1 is 0.91 using LR when $N_{\mathrm{G}} = 150$k and $d_{\mathrm{f}} = 50$k.

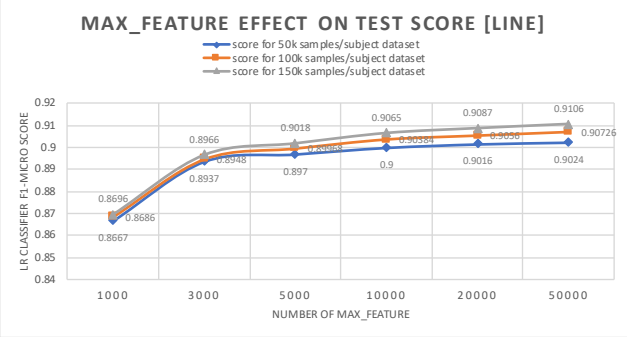

Fig. 2. Performance changes with d_{f} for LR.

B. Multilayer Perceptron

In the preliminary study, we build a typical Multilayer Perceptron (MLP) with 3 hidden layers, each having 1024, 512, and 512 neurons, respectively (ReLU Activation). The input layer contains 5000 neurons and the output layer contains 6 neurons (softmax activation), corresponding to 6 SCs. We also use a dropout fraction of 0.2 to prevent the model from overfitting. *In this setting, documents are classified into 6 SCs including the Others category*. At first, we pre-process each abstract in a similar way as the supervised learning model by applying tokenization, lemmatization, and stopword removal. We then represent each abstract using a TF-IDF feature vector, the elements of which are the top d_{f} tokens ranked by their TF-IDF scores. For comparison, we run LR, RF, MNB, and SVM models on the same dataset. In these experiments, we split the ground truth so the training set takes 90%, and the testing set takes 10%.

We considered representing abstracts using word embedding (WE). We first tried GloVe trained on 6B tokens to encode tokens. The best F1 obtained was less than 0.80. Analysis indicates the vocabulary of GloVe we used has only 37% overlap on average with the vocabulary of WoS abstracts. We then employed Word2Vec Skip-Gram model with softmax to generate our own word embedding representation. The F1 was even lower. We believe this was because without the TF-IDF feature, Word2vec gives equal importance to each word, which results in picking trivial words for vector representation. In the future, we will consider WE on TF-IDF ranked tokens.

The results tabulated in Table I indicate that (1) adding the Others category significantly decreases the performance given the same vector dimension and training data; (2) Given a sufficiently large training corpus, MLP achieves comparable performance to traditional supervised models (LR and RF). The contingency matrix indicates that the best F1 is achieved for CMPSC (94%) and the lowest F1 is achieved for the Others category (65%).

TABLE I
PERFORMANCE OF MLP COMPARED WITH OTHER CLASSIFIERS.

Metric	LR	RF	MNB	SVM	MLP
micro-F1	0.83	0.83	0.78	0.82	0.83
T_{test} (sec)	4.78	8.67	5.40	6.74	6.16

[1] T_{test}: time spent on testing 90k samples. All models are trained with 6 SCs, each containing $N_{\mathrm{G}} = 150$k samples. Each abstract is represented by a vector of $d_{\mathrm{f}} = 5000$.
[2] LR, RF, MNB, and SVM are trained on a server with 32 logical cores and 315GB RAM; MLP is trained on a GPU server with NVIDEA GTX 1080 Ti and 64GB RAM.

Using the fastest model LR, we classified 3 randomly selected sets from CiteSeerX, each containing 1M documents. The macro-average percentage of each SC is below: PHYS (11.35%), CHEM (12.37%), BIO (18.62%), MATSC (5.35%), CMPSC (7.58%), and Others (44.73%).

IV. CONCLUSIONS

Using a combination of machine learning and information retrieval methods, we cleansed metadata of 4.2M academic documents in CiteSeerX, and designed a model to classify academic documents into 6 SCs. Future could investigate the characteristics of the remaining 5.8M unmatched documents and classify the entire dataset into 252 SCs using the best trained model on the WoS. One could also investigate the approach of assigning multiple SCs to a document.

REFERENCES

[1] C. Caragea, J. Wu, A. Ciobanu, K. Williams, J. Fernández-Ramírez, H.-H. Chen, Z. Wu, and L. Giles. *CiteSeerX: A Scholarly Big Dataset*, pages 311–322. Springer International Publishing, Cham, 2014.

[2] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines. In *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '03, pages 37–48, 2003.

[3] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 42–49, New York, NY, USA, 2004. ACM.

[4] A. Sefid, J. Wu, A. C. Ge, J. Zhao, L. Liu, C. Caragea, P. Mitra, and C. L. Giles. Cleaning noisy and heterogeneous metadata for record linking across scholarly big datasets. In *Proceedings of the Thirty-First AAAI Innovative Applications of Artificial Intelligence Conference, January 29 -31, 2019, Honolulu, Hawaii, USA.*, Accepted.

[5] J. Wu, J. Killian, H. Yang, K. Williams, S. R. Choudhury, S. Tuarob, C. Caragea, and C. L. Giles. 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, pages 13:1–13:8, New York, NY, USA, 2015. ACM.

[6] J. Wu, C. Liang, H. Yang, and C. L. Giles. Citeseerx data: Semanticizing scholarly papers. In *Proceedings of the International Workshop on Semantic Big Data*, SBD '16, pages 2:1–2:6, New York, NY, USA, 2016. ACM.

[7] J. Wu, K. Williams, H. Chen, M. Khabsa, C. Caragea, A. Ororbia, D. Jordan, and C. L. Giles. 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.*, pages 2930–2937, 2014.