Tablepedia: Automating PDF Table Reading in an Experimental Evidence Exploration and Analytic System

Wenhao Yu
University of Notre Dame
Notre Dame, IN
wenhaoyu97@gmail.com

Zongze Li
Sichuan University
Chengdu, Sichuan, China
waterobrien@gmail.com

Qingkai Zeng
University of Notre Dame
Notre Dame, IN
qzeng@nd.edu

Meng Jiang
University of Notre Dame
Notre Dame, IN
mjiang2@nd.edu

ABSTRACT

Web research, data science, and artificial intelligence have been rapidly changing our life and society. Researchers and practitioners in the fields take a large amount of time to read literature and compare existing approaches. It would significantly improve their efficiency if there was a system that extracted and managed experimental evidences (say, a specific method achieves a score of a specific metric on a specific dataset) from tables of paper PDFs for search, exploration, and analytic. We build such a demonstration system, called *Tablepedia*, that use rule-based and learning-based methods to automate the "reading" of PDF tables. It has three modules: template recognition, unification, and SQL operations. We implement three functions to facilitate research and practice: (1) finding related methods and datasets, (2) finding top-performing baseline methods, and (3) finding conflicting reported numbers. A pointer to a screencast on Vimeo: https://vimeo.com/310162310

KEYWORDS

PDF table, Experimental evidence, Data science

ACM Reference Format:
Wenhao Yu, Zongze Li, Qingkai Zeng, and Meng Jiang. 2019. Tablepedia: Automating PDF Table Reading in an Experimental Evidence Exploration and Analytic System. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3308558.3314118

1 INTRODUCTION

The motivation of building the proposed system was originally from our literature study on *multilabel classification* which is to predict the labels of objects where multiple labels may be assigned to each object. It cost us as long as 23 days to collect, read, and digest hundreds of related works. We found two papers of this topic that were accepted to ACM SIGKDD 2017 Research Track: PPDSparse [11] and AnneXML [9]. Each of them proposed a new multilabel classification model and compared with baseline methods. They both reproduced and tested existing methods (such as SLEEC, FastXML, PfastreXML, and PDSparse) on publicly available data sets (such as AmazonCat-13K, Delicious-200K, and WikiLSHTC-325K) using standard evaluation metrics (such as Precision@1, P@3, and P@5). Table 1 summarizes and compares the numbers given by the two papers, [11] on the left and [9] on the right. We find out that almost half of the pairs have bigger than 3% difference on the scores, which has been able to be claimed as significant improvement on well-accepted benchmarks. This may be due to the random initialization, parameter settings, or computational environments. We have no idea about the true reason, but we argue that it is worthwhile of investigating the *experimental evidences* in data science literature.

Dataset	(%)	SLEEC	FastXML	PfastreXML	PDSparse
AmazonCat	P@1	90.56/89.19	94.02/93.10	86.06/89.94	87.43/89.31
-13K	P@3	76.96/75.17	79.93/78.18	86.06/77.24	87.43/74.03
	P@5	62.63/61.09	64.90/63.38	63.65/63.53	56.70/60.11
Delicious	P@1	47.78/47.03	48.85/43.20	26.66/37.62	37.69/34.37
-200K	P@3	42.05/41.67	42.84/38.68	23.56/35.62	30.16/29.48
	P@5	39.29/38.88	39.83/36.21	23.21/34.03	27.01/27.04
WikiLSHTC	P@1	58.34/55.57	50.01/49.75	57.17/58.10	60.70/61.26
-325K	P@3	36.70/33.06	32.83/33.10	37.03/37.61	39.62/39.48
	P@5	26.45/24.07	24.13/24.45	27.19/27.69	29.20/28.79

Table 1: Tablepedia found inconsistent numbers by two KDD papers [11] (left) and [9] (right) for multilabel classification. Precision differences of bigger than 3% are underlined. It is worthwhile of attention to the inconsistency.

Again, it took us long long time to make the comparison table (and several other tables that compared methods in other conference or journal papers). Therefore, we aim at building a system to extract and manage such experimental results in the literature. We hope that researchers and practitioners in the fields of data science and artificial intelligence will use it as Wikipedia to satisfy their needs of exploring and analyzing the experimental evidences.

The key challenges lie in automating the "reading" of tables in the experimental sections of paper PDFs. *First*, there was no well-defined structure of experimental evidence. The tables are embedded in the PDF format. It takes careful engineering efforts on cropping, parsing, and cleaning the tables. *Second*, the tables have different kinds of templates, so there was no standard of interpreting the cells. *Third*, the roles of row and column names (such as SLEEC and P@1), say, datasets or methods or metrics, are unknown. The gap between PDF table and queryable database is huge.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3314118

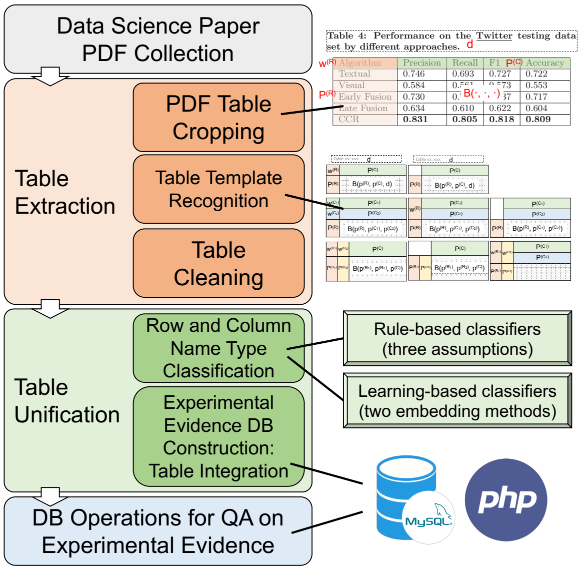

Data Science Paper PDF Collection
Table Extraction
PDF Table Cropping
Table Template Recognition
Table Cleaning
Table 4: Performance on the Twitter testing data set by different approaches.
d
w(R)
P(R)
P(C)
B(·, ·, ·)
Algorithm | Precision | Recall | F1 | Accuracy
Textual | 0.746 | 0.693 | 0.727 | 0.722
Visual | 0.584 | 0.561 | 0.573 | 0.553
Early Fusion | 0.730 | 0. | 37 | 0.717
Late Fusion | 0.634 | 0.610 | 0.622 | 0.604
CCR | 0.831 | 0.805 | 0.818 | 0.809
Table xx: xxx
B(p(R), p(C), d)
B(p(R), p(C1), p(C2))
B(p(R1), p(R2), p(C))
Table Unification
Row and Column Name Type Classification
Experimental Evidence DB Construction: Table Integration
Rule-based classifiers (three assumptions)
Learning-based classifiers (two embedding methods)
MySQL
php
DB Operations for QA on Experimental Evidence

Figure 1: Tablepedia workflow: from PDF collection, to table extraction, to experimental evidence database construction, to database operations and visualization.

Proposed approach. This paper presents a novel system, called Tablepedia, which transforms data science paper PDFs into a structured database of experimental evidences, and support multiple exploratory and analytic functions over the constructed database for knowledge discovery. It has three modules. The first module *table extraction* crops the tables from PDFs and recognize their templates. The second module *table unification* classifies the column names and row names into the three types of labels (method, dataset, and metric) and then unifies each cell into a **(method, dataset, metric, score, source)-tuple**. The score is the cell's value and the source is the PDF file name, page, and number of the table. This module constructs a five-column database of the tuples for every table that contains experimental results. The third module *database operation for QA* uses SQL operations (i.e., *select* and *join*) for question-answering on the experimental result database.

Contributions. The contributions and features of the Tablepedia system are summarized as follows.

- A novel system that extracts experimental evidences from massive literature in PDF format. This builds up the first experimental knowledge base for data science and artificial intelligence research.
- An effort-light framework that leverages both rule-based and learning-based methods to unify the tables of experimental results into (method, dataset, metric, score, source)-tuples.
- Capabilities for exploration and analysis over the structured knowledge base to facilitate research and practice.

The Tablepedia demo system will be made available online for interactive use after its demonstration in the conference. A pointer to a screencast on Vimeo: https://vimeo.com/310052262

2 THE TABLEPEDIA SYSTEM

In this section, we first introduce the workflow of our Tablepedia system and the details of the three modules of the system.

Overview. Figure 1 shows the overflow. Tablepedia collects a set of data science paper PDFs. It has three modules to process the PDF data. It first crops the tables from PDFs, recognizes the table templates, and cleans the table data. Second, it classifies the row and column names of each table into three categories (method, dataset, metric). The experimental evidence database is constructed through the integration of table cells. Lastly, it designs database operations for knowledge exploration in the structured database.

Expected output and impact. Figure 2 shows a snapshot of the experimental evidence database. It has several examples of data records. They are experimental facts that can be found in tables of conference and journal papers on building recommender systems that were published in the same year: TOIS'11 [8], TIST'11 [6], and WSDM'11 [7]. The tables share popular method names such as "User Mean", "NMF", and "PMF". Which method performs the best? Are the reported numbers of their performances consistent in these tables? When the tables were well structured into such a database, the above questions could be easily answered. The number of publications in the field of data science has been tremendously increasing because of the great use of data mining and machine learning in real applications. Practitioners are curious about what method will generate good performance on a specific task and dataset. Researchers are wondering whether the baseline methods are the state-of-the-art and whether the reported numbers on the baselines are correct when they review papers.

2.1 Table Extraction

We use Tabula to extract tabular content from PDF [2]. Tabula was created by Manuel Aristaran *et al.* with the first release made available early 2013 as an open source project. The developers stated that they were inspired by academic papers [12] about analysis and extraction of tabular content. Tabula is available as a Java library [1]. Unfortunately, it does not work for scanned documents, so we filter those files out.

A table $T = \{\mathcal{R}, \mathcal{C}, d, \mathcal{B}\}$ has four components: (1) a model of horizontal *Rows* (identifiable by name) $\mathcal{R}$, (2) a model of horizontal *Columns* $\mathcal{C}$, (3) *Caption* and the set of words in the caption d, and (4) cells (data elements) in the table's *Body* $\mathcal{B}$. We observe that the tables can be categorized into eight major templates (with very few exceptions). Figure 3 visualizes the components of each template.

For cleaning the raw tables, we count the number of digit-format cells in the table: we filter out the tables that have fewer than 6 digit cells, and thus, we get rid of more than 99% of the non-experimental result tables. We use [10] to remove the text that was not the table's caption but located around the table.

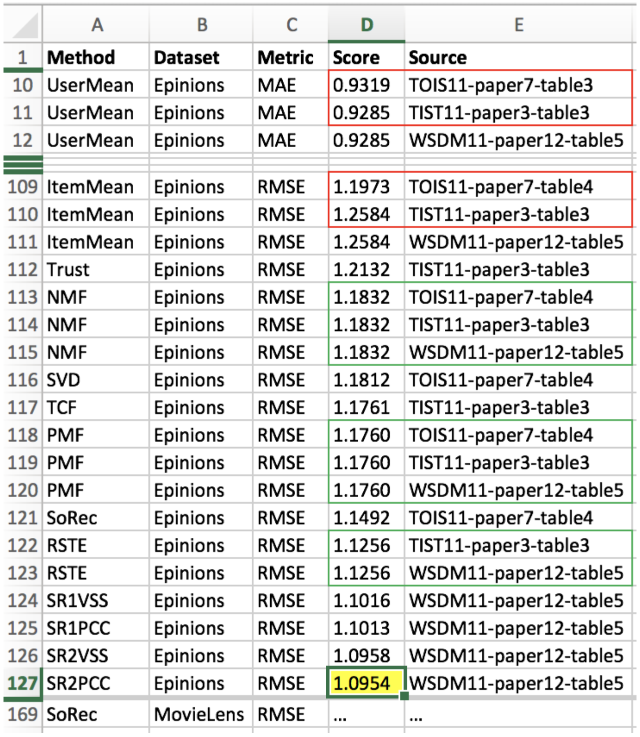
2.2 Table Unification

We define the set of concept items that can be found as row names or column names:

$$\mathcal{P} = \cup_{T=[\mathcal{R},\mathcal{C},d,\mathcal{B}]} P^{(R_{(\cdot)})} \cup P^{(C_{(\cdot)})}, \tag{1}$$

where T is a table, $P^{(R_{(\cdot)})}$ is the set of row names (no matter single row or double rows), and $P^{(C_{(\cdot)})}$ is the set of column names. We denote by $\mathcal{L}$ by the set of three labels for the concept items:

$$\mathcal{L} = \{\text{“}method\text{”}, \text{“}dataset\text{”}, \text{“}metric\text{”}\}. \tag{2}$$



	A	B	C	D	E
1	Method	Dataset	Metric	Score	Source
10	UserMean	Epinions	MAE	0.9319	TOIS11-paper7-table3
11	UserMean	Epinions	MAE	0.9285	TIST11-paper3-table3
12	UserMean	Epinions	MAE	0.9285	WSDM11-paper12-table5
109	ItemMean	Epinions	RMSE	1.1973	TOIS11-paper7-table4
110	ItemMean	Epinions	RMSE	1.2584	TIST11-paper3-table3
111	ItemMean	Epinions	RMSE	1.2584	WSDM11-paper12-table5
112	Trust	Epinions	RMSE	1.2132	TIST11-paper3-table3
113	NMF	Epinions	RMSE	1.1832	TOIS11-paper7-table4
114	NMF	Epinions	RMSE	1.1832	TIST11-paper3-table3
115	NMF	Epinions	RMSE	1.1832	WSDM11-paper12-table5
116	SVD	Epinions	RMSE	1.1812	TOIS11-paper7-table4
117	TCF	Epinions	RMSE	1.1761	TIST11-paper3-table3
118	PMF	Epinions	RMSE	1.1760	TOIS11-paper7-table4
119	PMF	Epinions	RMSE	1.1760	TIST11-paper3-table3
120	PMF	Epinions	RMSE	1.1760	WSDM11-paper12-table5
121	SoRec	Epinions	RMSE	1.1492	TOIS11-paper7-table4
122	RSTE	Epinions	RMSE	1.1256	TIST11-paper3-table3
123	RSTE	Epinions	RMSE	1.1256	WSDM11-paper12-table5
124	SR1VSS	Epinions	RMSE	1.1016	WSDM11-paper12-table5
125	SR1PCC	Epinions	RMSE	1.1013	WSDM11-paper12-table5
126	SR2VSS	Epinions	RMSE	1.0958	WSDM11-paper12-table5
127	SR2PCC	Epinions	RMSE	1.0954	WSDM11-paper12-table5
169	SoRec	MovieLens	RMSE	...	...

Figure 2: Tablepedia generates this experimental evidence database from data science paper PDFs. For a dataset and an evaluation metric, one can use the database to check what the state-of-the-art (highlighted in yellow) is and whether the reported numbers in existing researches are consistent (green box) or conflicting (red box).

Then we define table unification as a two-step problem.

Problem (Table unification). ***Given*** *a set of tables* $\{T\}$ *and each table has been well defined based on its template, (1)* ***classify*** *the concepts into three categories, or say,* ***find*** *a classification function* $f: \mathcal{P} \rightarrow \mathcal{L}$*; (2)* ***unify*** *the cells into (method, dataset, metric, score, source)-tuples, or say,* ***find*** *a function of three variables* g*:* $P^{(\text{“}method\text{”})} \times P^{(\text{“}dataset\text{”})} \times P^{(\text{“}metric\text{”})} \rightarrow \mathbb{R}$*, where the target value is the score (a real number) as in the Table's body function B.*

Tablepedia develops an ensemble learning approach that iteratively predicts the labels of concept items using two classifiers of different methodologies. The first classifier is an **assumption/rule-based** method. The first assumption is:

Assumption 1 (Row/column header indication). *If the upper-leftmost cell of the table has a specific word (e.g., "Methods", "Dataset"), the names on the corresponding columns/rows are more likely to have the label as the word indicates.*

For example, if the upper top cell of a table has word "Methods", then the row names such as "User Mean", "Item Mean", and "NMF" are likely to be labelled as "method". Then we use these "seed" concept items to label columns, rows, and captions of all the tables. If the columns/rows/captions are partially labelled, we will be able to use the following assumptions to predict the labels of the remaining concepts on the columns/rows/captions:

Assumption 2 (Row/column type consistency). *The concept items on the same column or row are likely to have the same type of label. For example, if we know (1) "NMF" is a "method" and (2) "SVD" locates in the same column/row as "NMF" does, then "SVD" is likely to be a "method".*

Assumption 3 (Cell context completeness). *A table often covers all the three types of labels on its columns, rows, and caption, in order to provide complete contexts [4] to explain the values in the cells. For example, if the caption has a metric name (i.e., "MAE") and the row names are methods, then the column names are likely to be datasets.*

The second classifier is a **learning-based** method. It learns low-dimensional representations (or called *embeddings*) of the concept items for label prediction. It learns (a) *semantic concept embeddings* [5] from the unstructured paper texts and (b) *structural concept embeddings* [3] from the co-occurrences of concepts in the table structures (i.e., columns and rows), and then feeds the concatenated feature vectors into standard multi-class classification models (e.g., random forests, support vector machines, neural networks) to predict the labels. This solution will be likely to assign the same label to different forms of the same concept.

The ensemble learning approach uses the boosting strategy to iteratively learn the classifiers. Relying too much on one classifier may result in lots of errors when the prediction confidence is weak. In each iteration, we only expand the set of classified concepts for next round of training by a certain small number. So after several iterations, we can have precise labels of the concept items.

The next step is to unify the table cells into the tuples and put into the five-column experimental result database (ERD). The ERD's quality relies on the accuracy of concept typing. Tablepedia achieves an F1 score of 0.8477, which is much higher than using rule-based or learning-based method only (0.6542).

2.3 Database Operations for QA

When the ERD was constructed, we would be able to use SQL queries and operations to answer interesting questions. There could be many questions and corresponding SQL queries.

Question 1. How many methods were used/proposed on the Epinions dataset? And how many metrics were used?

Question 2. What are the top three methods on the Epinions dataset if the evaluation metric is RMSE?

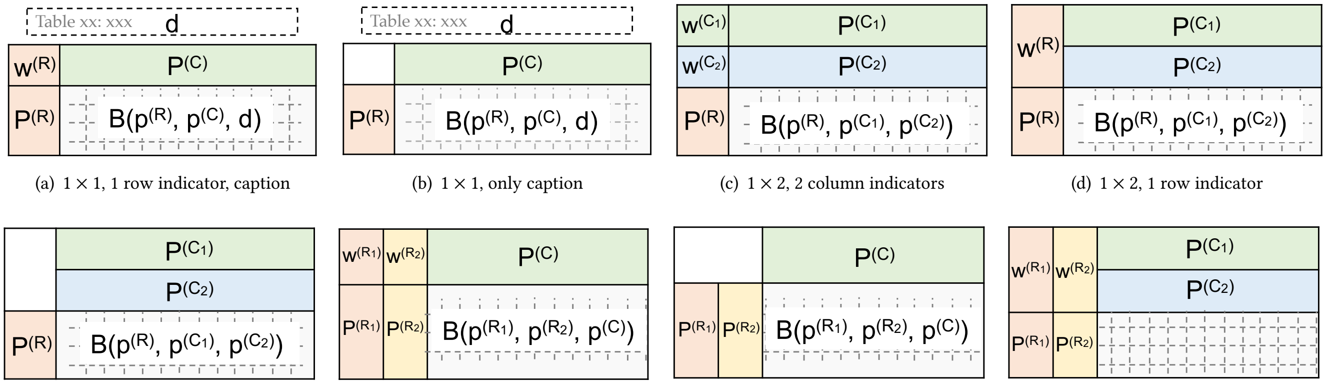
Question 3. Are there conflicting reported numbers in the database? What are they?

SQL consists of many types of expressions, predicates and statements such as *select*, *join*, and *distinct*, based upon *relational algebra* and *tuple relational calculus*. Suppose the experimental result data table is constructed and named as "ERD". Here are the SQL queries that find answers to the above questions.

SQL Queries 1.
select count(distinct Method) from ERD where Dataset="Epinions";
select count(distinct Metric) from ERD where Dataset="Epinions";

SQL Query 2. select * from ERD where Dataset = "Epinions" and Metric = "RMSE" order by Score desc limit 3;

SQL Query 3. select distinct d1.Method, d1.Dataset, d1.Metric, d1.Score, d1.Source from ERD as d1, ERD as d2 where d1.Method = d2.Method and d1.Dataset = d2.Dataset and d1.Metric = d2.Metric and d1.Score <> d2.Score order by d1.Method, d1.Dataset, d1.Metric;



(a) 1 × 1, 1 row indicator, caption (b) 1 × 1, only caption (c) 1 × 2, 2 column indicators (d) 1 × 2, 1 row indicator

(e) 1 × 2, no indicator (f) 2 × 1, 2 row indicators (g) 2 × 1, no indicator (h) 2 × 2, 2 row/column indicators

Figure 3: Eight major table templates: We will use the first seven templates which cover more than 95% of the tables in our dataset. The cells in the table's body are triplets based on rows/columns/caption. (Best view in color)

The term count is used for question "how many"; order by is used for ranking/finding "top three"; and the third query uses *self-join* to compare values in a column ("Score") with other values in the same column in the same table ("ERD").

We developed user-friendly functions in this module to answer the questions. For example, users can fill in the underlined values in the questions, the SQL queries will be updated, and then correct answers will be returned.

3 DATA STATISTICS

We downloaded from web portals such as ACM Digital Libraries a PDF file collection of four data science conference proceedings (WWW, SIGKDD, ICDM, and WSDM) and three ACM transactions (TOIS, TIST, and TKDD) last decade (2008–2017). After careful PDF converting, cropping, and cleaning, we have **456 tables**.

Tablepedia categorized **4,476 concepts** in the 456 tables into three classes, {dataset, method, metric}. The resulting database has as many as **29,081 data records** (or called experimental result facts). The database includes **1,541** unique datasets, **1,685** unique methods, and **450** unique metric names. Each dataset, method, and metric has **18.9**, **17.3**, and **64.6** related data records in average, respectively. The associations between the concepts are rich.

4 DEMO SCENARIOS

Tablepedia uses the database to answer the following questions. This is just to show the power of exploring quantitative knowledge in the experimental result database and the usefulness of our approach. Because the database was constructed with only 456 tables, we are **NOT** claiming that the answers to these questions are the truths all over the tons of literature.

***Question 1:* Find related methods, metrics, and datasets.**

Q-1(a) How many methods were used for the Epinions dataset?
select count(distinct Method) from ERD where Dataset="Epinions";
A-1(a) 36. If one uses more SQL queries to look for the detail, one will see the method names such as "UserMean", "ItemMean", "Trust", "NMF", "SVD", "TCF", "PMF", "SoRec", and "RSTE".

Q1(b) How many metrics were used to evaluate on Epinions?
select count(distinct Metric) from ERD where Dataset="Epinions";
A-1(b) 7. More queries will find the concrete metric names such as "F1 score", "Precision", "Recall", "MAE", and "RMSE".

Q1(c) How many datasets used with Epinions in the same table?
select count(distinct Dataset) from ERD where Source=(select (distinct Source) from ERD where Dataset= "Epinions");
A-1(c) 17. The data names are "Amazon", "Ciao", "Douban", and so on. They are popular datasets for evaluating recommender systems.

***Question 2:* Find top-performing methods on a dataset.**

Q2(a) What are the top 3 methods on Epinions in terms of RMSE?
select Method, Score from ERD where Dataset = "Epinions" and Metric = "RMSE" order by Score desc limit 3; // desc is for the fact that a smaller RMSE means a better performance.
A-2(a) "SR2pcc" (1.0954), "SR2vss" (1.0958), "SR1pcc" (1.1013).

Q2(b) What are the top 3 methods on Amazon in terms of F1?
select Method, Score from ERD where Dataset = "Amazon" and Metric = "F1" order by Score limit 3; // Compared to Q2(a), desc was deleted because a bigger F1 means a better performance.
A-2(b) "LEMON" (0.953), "LEMON-auto" (0.91), "LC" (0.815).

***Question 3:* Find conflicting reported numbers.**
Surprisingly, we found a large set of conflicting records in the database. A number of them are worthy of investigation: *First*, as presented in Table 1, the two KDD 2017 papers on multilabel classification, [11] and [9], gave different numbers for the same set of methods, the same datasets, and the same metrics, respectively. Though variance could happen when reproducing the results, we found many of the precision differences are bigger than 3%, which is often a sufficient margin to claim a new achievement! *Second*, if the dataset is Epinions, *plus* the metric is MAE, *plus* the ratio of training data is 80%, then we have three pairs of conflicting numbers reported by [8] and [6]: (1) UserMean: 0.9319 vs 0.9285, (2) ItemMean: 0.9115 vs 0.9913, (3) Trust: 0.9044 vs 0.9215. The same pairs can be observed for RMSE as well. *Finally*, we also find a number of conflicting pairs that were not correctly aligned because of the missing contexts in the extraction such as the ratio of training data and the number of dimensions. In this demo, while showing the importance of integrating PDF tables, we are aware of tons of challenging and interesting future works.

REFERENCES

[1] M. Aristarán. 2013. Tabula-Java. https://github.com/tabulapdf/tabula-java. (2013).

[2] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. 2012. A methodology for evaluating algorithms for table understanding in PDF documents. In *Proceedings of the 2012 ACM symposium on Document engineering*. ACM, 45–48.

[3] Huan Gui, Jialu Liu, Fangbo Tao, Meng Jiang, Brandon Norick, Lance Kaplan, and Jiawei Han. 2017. Embedding learning with events in heterogeneous information networks. *IEEE TKDE* 29, 11 (2017), 2428–2441.

[4] Meng Jiang, Christos Faloutsos, and Jiawei Han. 2016. Catchtartan: Representing and summarizing dynamic multicontextual behaviors. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 945–954.

[5] Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M Kaplan, Timothy P Hanratty, and Jiawei Han. 2017. Metapad: Meta pattern discovery from massive text corpora. In *KDD*. ACM, 877–886.

[6] Hao Ma, Irwin King, and Michael R Lyu. 2011. Learning to recommend with explicit and implicit social relations. *ACM TIST* 2, 3 (2011), 29.

[7] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM*. 287–296.

[8] Hao Ma, Tom Chao Zhou, Michael R Lyu, and Irwin King. 2011. Improving recommender systems by incorporating social contextual information. *ACM TOIS* 29, 2 (2011), 9.

[9] Yukihiro Tagami. 2017. AnnexML: Approximate nearest neighbor search for extreme multi-label classification. In *KDD*. 455–464.

[10] Yingchen Yang and Wo-Shun Luk. 2002. A framework for web table mining. In *Proceedings of the 4th international workshop on Web information and data management*. 36–42.

[11] Ian EH Yen, Xiangru Huang, Wei Dai, Pradeep Ravikumar, Inderjit Dhillon, and Eric Xing. 2017. Ppdsparse: A parallel primal-dual sparse method for extreme classification. In *KDD*. 545–553.

[12] Burcu Yildiz, Katharina Kaiser, and Silvia Miksch. 2005. pdf2table: A method to extract table information from pdf files. In *IICAI*. 1773–1785.