

# Project Instruction

## “Data Science Research Bot” (a.k.a. SciBot)

### Data Science CSE 40647/60647

Last updated: Sept. 5, 2017 (being updated)

#### Professor-in-charge:

Dr. Meng Jiang, [mjiang2@nd.edu](mailto:mjiang2@nd.edu)

Office: 326C Cushing Hall

Phone: (574) 631-7454

#### Teaching Assistant (TA):

Qi Li, [qli8@nd.edu](mailto:qli8@nd.edu)

Office: 212 Cushing Hall

#### Project goal:

Individual project, NOT group project.

On a large real-world dataset, students should be able to:

- Process raw data: data cleaning, data integration, data reduction, dimension reduction
- Describe data warehouse, OLAP, data cube concepts and technology that work on multi-dimensional data
- Use Apriori and FP-Growth for frequent pattern mining
- Describe diverse patterns, sequential patterns, graph patterns
- Use Decision Tree, Naïve Bayes, Ensembles for classification
- Describe SVMs and Neural Networks for classification
- Use K-Partitioning Methods (K-Means, etc.) for clustering
- Describe Kernel-based Clustering and Density-based Clustering
- Use appropriate measures to evaluate results of different functionalities

Students are required to accomplish tasks that will be described as “required tasks” below. Students are encouraged to do more tasks as either the recommended ones or the ones they like to do. Basically, the ultimate goal is to enrich the functionalities of the “SciBot” using data science and technology.

One example of the functionalities could be:

>> *What problem do you want to find methods that are strongly associated with?*

>> *(by user) document classification*

>> *The methods that are associated with the problem “document classification” are:*

*support\_vector\_machines (relative support: 0.37, confidence: 0.25)*

*decision\_tree (relative support: 0.32, confidence: 0.21)*

...

Students are also required to write a project report/ paper to describe their achievement including the following points *for each task*: (1) Motivation and task definition, (2) Approach, (3) Results, and (4) Discussions.

**Grading policy: (25% of the final score)**

Students are required to submit their code package + “readme” (.ZIP) and project report/paper (.PDF). There is no paper template requirement.

Students will volunteer to present their SciBot (tech and results) in two lectures. Classmates and the instructor will grade them based on the presentation. For the students who do not present, the instructor will grade their projects after all the lectures end. Note that we will have comparative grading – finishing all the required tasks cannot make sure that you have all the points.

Students are encouraged to implement algorithms such as Apriori, FP-Growth, and K-Means Clustering by themselves instead of calling Python packages.

Students are also encouraged to use Python packages (e.g., numpy and scipy) when they use advanced techniques (e.g., SVMs, Neural Networks, word2vec) to address challenging tasks.

Students are encouraged to compare different methods on the same task and discuss their advantages and disadvantages. Reasoning is always welcome in the paper.

Students are encouraged to share any annotation data (e.g., labels, hand-crafted rules) but not any segment of codes.

Students are encouraged to make a GUI for the SciBot. They are also encouraged to give a better name to their bots than “SciBot”.

Graders should have higher expectations on graduates than undergraduates – not only on the project results (more tasks, better performances) but also on writing (a workshop-quality paper of strong reasoning). Undergraduates will be applied with a uniform grading policy no matter what majors they have.

The project due is Nov 30, 2017. There will be NO extension. Significant updates are welcome before the final exam – students can send the updates to the instructor after the due by e-mail but they have to submit one version before the due.

**Academic Dishonesty:**

- The CSE and du lac honor code will be strictly followed.
- All assignments are individual unless instructed. You can discuss the assignment at a high level, but you should independently and individually write down the answers and/or the program. The sharing and copying of homework solutions or programs or functions or exams will be considered cheating.
- All the references and sources should be carefully provided and cited.
- Entering Notre Dame you were required to study the on-line edition of the Academic Code of Honor, to pass a quiz on it, and to sign a pledge to abide by it. The full Code and a Student Guide to the Academic code of Honor are available at: <http://honorcode.nd.edu>.
- Perhaps the most fundamental sentence is the beginning of section IV-B: “The pledge to uphold the Academic Code of Honor includes an understanding that a

student's submitted work, graded or ungraded – examinations, draft copies, papers, homework assignments, extra credit work, etc. - must be his or her own."

### Dataset introduction:

The dataset has both structured and unstructured information of over five thousand data science research papers. It includes three zip files:

1. pdf.zip (4.7GB; unzip ~5.5 GB): raw unstructured data (actually you don't have to use this huge file)

<https://www.dropbox.com/s/460h772tpuceew5/pdf.zip?dl=0>

It has 64 folders/proceedings. Each folder is named as "[CONF][YEAR]":

- CONF: {icdm, kdd, wsdm, www}  
icdm: IEEE International Conference on Data Mining  
kdd: ACM SIGKDD Conference on Knowledge Discovery and Data Mining  
wsdm: ACM Conference on Web Search and Data Mining  
www: International Conference on World Wide Web
- YEAR: {94, 95, ..., 99, 00, 01, ..., 16}  
from 1994 to 2016

Each folder has an incomplete set of papers of the proceeding of CONF-YEAR. The papers are named as "[PDFID].pdf":

- PDFID: {icdm01-d0, ...}

2. text.zip (~95MB; unzip ~270MB): raw unstructured data

<https://www.dropbox.com/s/o0f7qjb5mobmfvt/text.zip?dl=0>

It has the same folder names and file names as pdf.zip. The only difference is the files' ext. name (".txt" here, ".pdf" in pdf.zip). A Python package was used to transfer \*.pdf into \*.txt, but the text looks incomplete and noisy.

- Practitioners are recommended to skip the REFERENCE section when they mine knowledge from the text data.

3. microsoft.zip (~24MB; unzip ~100MB): raw structured data

<https://www.dropbox.com/s/o9qzhbdd0pmk5wm/microsoft.zip?dl=0>

It has five files. All except "index.txt" were provided by Microsoft Academic Search (MAS) engine. "index.txt" was created by the instructor to bridge the structured and unstructured data with entry id (PDFID and PID).

(1) index.txt

Folder name in pdf.zip / txt.zip	PDFID (file name) in pdf.zip (*.pdf) / txt.zip (*.txt): (paper id in PDFs)	PID (paper id in MAS database)	TITLE (lower case)
----------------------------------	--	--------------------------------	--------------------

(2) Papers.txt

PID (paper id in MAS database)
TITLE_CASE (case sensitive)
TITLE (lower case)
YEAR (year of proceeding)
DATE_OF_PROCEEDING (not recommended to use)

DOI (not recommended to use)
CONF_FULL_NAME (not recommended to use)
CONF (abbreviation, lower case)*
N/A
CID (conference id; not proceeding id!)**
N/A

\* The dataset is noisy. NOT every entry can be correlated across the files, for example, CONF has conference names that are not included in {icdm, kdd, wsdm, www}.

\*\* One-to-one mapping between CONF and CID.

### (3) PaperKeywords.txt

PID (paper id in MAS database)
KEYWORD (keyword in lower case)*
KID (keyword id; not recommended to use)

\* It has a very limited set of keywords. We will process the text data for more structured semantic information of the papers.

### (4) PaperAuthorAffiliation.txt

PID (paper id in MAS database)
AID (author id in MAS database)
FID (affiliation id in MAS database)
AFF_ORG (original affiliation name, not recommended to use)
AFF (normalized affiliation name)*
SID (author sequence number: "1" = the first author, "3" = the 3 <sup>rd</sup> author)**

\* One-to-one mapping between AFF and FID.

\*\* The author information of a paper may not be complete. It may only have the 1<sup>st</sup>, 2<sup>nd</sup>, and 4<sup>th</sup> authors.

### (5) Authors.txt

AID (authored in MAS database)	AUT (author name in lower case)
--------------------------------	---------------------------------

## Required tasks:

### Task 1: Data preprocessing

**Q1:** Given the above files from multiple sources, can we integrate the data, clean the data (work with incomplete/missing entries and redundancy/unnecessary entries), describe the data using statistics and visualization (distributions, etc.)? What are the data objects and what are the attributes?

**Techs:** Data cleaning, data integration, data description, statistical analysis, data visualization.

### Hints:

1. PDFID-PID mapping in index.txt can be used to integrate paper text (txt.zip), Papers.txt, PaperKeywords.txt and PaperAuthorAffiliation.txt.
2. AID can be used to integrate PaperAuthorAffiliation.txt and Authors.txt.

PDFID	(PID)	CONF	YEAR	LIST_AUTHOR	LIST_KEYWORD	...	TEXT	(PDF)
-------	-------	------	------	-------------	--------------	-----	------	-------

--	--	--	--	--	--	--	--	--

## Task 2: Entity mining: Candidate generation and quality assessment

**Q2:** Given the text data, can we mine entities, e.g., “text categorization”, “document classification”, “naïve bayes”, “decision tree”, “support vector machines”, “SVM”, “SVMs”? Can we propose at least one measure of entity quality and rank them by it?

**Techs:** Frequent pattern mining, measures (outlier-ness like Z-score), hand-crafted rule matching

### Hints:

1. Entity names are a subset of words or phrases. Relational phrases or stop phrases are not entity names, e.g., “turn\_out\_to\_be”, “in\_this\_paper”.
2. Use the given keyword list as entity / phrase candidates. Suppose a document has 1,000,000 words. It has 1,000 “decision” and 1,000 “tree”. Assuming an even distribution of the words, we may only have one “decision tree”. The observed number could be much bigger.
3. Rules of lexical features are useful for generating entity candidates. For example, we may often see “... Support Vector Machines ...”, “support vector machines (SVMs)”, “... non-negative matrix factorization (NMF) ...” Then we use outlier-ness to evaluate the quality of the candidates.
4. We may generate N-grams (N=2,3,4...) as phrase / entity candidates. However, the number could be huge. Can we use heuristics (like the rules above) to generate a proper-sized set of candidates?
5. If we consider phrase / entity candidates as patterns (word itemsets), another possible quality measure is *absolute support* of the pattern (i.e., count of the pattern in the text data) if we consider each sentence / paragraph / paper as a transaction and words as items.
6. Label a set of quality entities. Evaluate the performance of different measures and different candidate sources / generation methods.

PDFID	LIST_ENTITY

## Task 3: Entity typing

**Q3:** Given quality entities and text data, can we assign types to the entities? Basically, we consider four major types: \$Problem, \$Method, \$Metric, \$Dataset.

\$Problem	text categorization, document classification, fraud detection ...
\$Method	naïve bayes, decision tree, support vector machines ...
\$Metric	accuracy, precision, recall, F1 score ...
\$Dataset	netflix, youtube, movielens, facebook, twitter, dblp ...

**Techs:** Measures (outlier-ness like Z-score), dimension reduction, classification

### Hints:

1. Take an entity as a data object. The attributes are contextual words around the entity in the text data. Suppose we have an N-size window and take each word in the window as an attribute. Then we can measure the probability of assigning a type to an entity. We assume that if the word “method”, “model” or “approach” has a high Z-score to appear in the context of an entity, the entity is likely to be typed as “\$Method”.
2. Here we carefully type entities. We want high accuracy but not good coverage. We use supervised methods (for classification – Naïve Bayes or decision tree or

others) and feed with the set of contextual words (attributes) to type other entities. If the number of attributes is too large, we can consider to use dimension reduction (like PCA or SVD).

PDFID	LIST_PROBLEM	LIST_METHOD	LIST_METRIC	LIST_DATASET

#### Task 4: Collaboration discovery

**Q4:** Given the paper-author data, find frequent author-sets (as patterns): which two/three/four authors often collaborate together?

**Techs:** Frequent pattern mining (Apriori, FP-Growth).

**Hints:**

1. Here each paper is considered as a transaction. Each author is an item.

#### Task 5: Problem-method association mining

**Q5:** Given the paper-problem-method data, find strong association rules, problem  $X \rightarrow$  method Y, or method  $X \rightarrow$  problem Y, of high support and confidence.

**Techs:** Association rule mining.

**Hints:**

1. Here each paper is considered as a transaction. Each problem/method is an item.

#### Task 6: Problem/method/author-to-conference classification

**Q6:** Given a problem/method/author, predict if a conference has papers of it.

**Techs:** Binary classification (Naïve Bayes, Decision Tree).

**Hints:**

1. What are the attributes (features) you want to use?
2. How to set up training and testing? Please evaluate the performance on different features, different models, and different setups.

#### Task 7: Paper clustering

**Q7:** Given a set of papers, cluster them into K groups.

**Techs:** K-partitioning clustering methods (K-Means).

**Hints:**

1. What are the attributes (features) you want to use?
2. Suppose  $K = 4$  and the ground-truth is the conference. Please evaluate the performance on different features and different methods.

### **Recommended tasks:**

#### Task 1+: Data preprocessing

**Q1+:** Study the data distributions: Do you find power-law, Poisson, or normal distributions between variables? Can you explain them?

**Techs:** Statistical analysis.

#### Task 2+: Entity mining: Candidate generation and quality assessment

**Q2+:** Can you use auxiliary sources (e.g., stop word list) or auxiliary criteria to further improve the quality of entity names you mined?

**Techs:** Classification (good entity name: "yes", "no").

#### Task 3+: Entity typing

**Q3+:** Can you use cluster analysis on the entities to type entities by clusters? Given what kind of features, the entities might be grouped together if they had the same type?  
**Techs:** Clustering

Task 4+: Advisor-advisee discovery

**Q4+:** Can you find advisor-advisee relations from collaborations?

**Techs:** Measures (like Kulc).

Task 8: Manage the data with data cube

**Q8:** Given enriched structured data, can we construct a data cube and compute iceberg cubes for query-based applications? E.g., expert recommendation: Given a problem, list authors, papers and other information that help related research.

**Techs:** Data cube, iceberg cube, closed cells, etc.

**Hints:**

1. Each paper is considered as a transaction. The cell maintains a set of papers. A paper may be in multiple cells. We count the size of paper set for the cube computation.
2. For a list of entities, the attribute types are dimensions (e.g., problem, method, dataset, author, conference); the attribute values are the dimension values (e.g., "naïve bayes", "decision tree").
3. More functionalities of the data cube, and efficiency analysis are welcome.

Task 9: Pattern-based entity recognition and typing

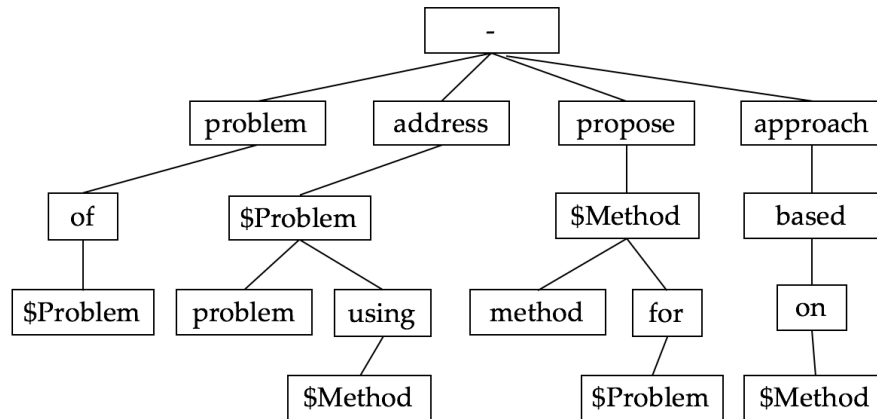
**Q9-1:** Given entity names, can we find frequent patterns around the entities? We replace concrete entity names as "\$Entity". You can find more entities via pattern matching.

**Q9-2:** Given seed typed entities (methods, problems, etc.), can we find concrete frequent patterns around the typed entities? We replace concrete method / problem entities as "\$Method" / "\$Problem". Those patterns indicate that you may be able to find more entities of the specific types.

**Techs:** Constraint-based frequent pattern mining.

**Hints:**

1. An iterative process that first generate and evaluate the support of patterns such as "problem of \$Problem", "address \$Problem problem", "propose \$Method method", "approach based on \$Method", "propose \$Method for \$Problem", "address \$Problem using \$Method" and then recognize more entities and their types by matching the patterns in the text and repeat until convergence.
2. How to generate the patterns *efficiently*? How to match the text with patterns *efficiently*? The data structure of Trie Tree (<https://en.wikipedia.org/wiki/Trie>) is recommended. Suppose we have the above 6 patterns. We can construct a tree below. It is easier to search in the tree than to match string patterns.



PDFID	LIST_PROBLEM	LIST_METHOD	LIST_METRIC	LIST_DATASET

#### Task 10: Problem/method/author clustering

**Q10:** Given a set of problems/methods/authors, cluster them into K groups. Evaluate the clustering results in a proper way.

**Techs:** K-partitioning clustering methods (K-Means).

#### Task 11: Attribute discovery

**Q11:** Suppose we use rules to type digit number as \$Digit. Can we find the size of datasets used in the papers? Can we find the performance of methods?

**Techs:** Constraint-based frequent pattern mining.

#### Task 12: Ensemble learning

**Q12:** Suppose we have multiple models/methods for a specific task (actually you do have if you've finished the required tasks). Can we use ensemble methods to further improve the performance?

**Techs:** Ensemble methods (bagging, Adaboost, etc.).

#### Task 13: Practice with advanced classification and clustering methods

**Q13:** Can you solve the above tasks with advanced classification models (e.g., SVMs, Neural Networks) and clustering methods (e.g., spectral clustering)?

#### Task 14: Other interesting tasks related to other data entries/attribute like "affiliation ranking on a specific method/problem".

#### Task 15: Data visualization is encouraged.

Examples: Project results from UIUC Summer 2017 Data mining course (10 weeks, 3 lectures per week, 5 written assignments).



## (1) A Web UI to manually label “problems”, “methods”, “metrics”, etc.

dd15-p1006  
dd15-p1015  
dd15-p1025  
dd15-p1035  
dd15-p1045  
dd15-p1055  
dd15-p1065  
dd15-p1075  
dd15-p1085  
dd15-p1095  
dd15-p1105  
dd15-p1115  
dd15-p1125  
dd15-p1135  
dd15-p1145  
dd15-p1155  
dd15-p1165  
dd15-p1175  
dd15-p1185  
dd15-p1195  
dd15-p1205  
dd15-p1215  
dd15-p1225  
dd15-p1235  
dd15-p1245  
dd15-p1255  
dd15-p1265  
dd15-p1275  
dd15-p1285  
dd15-p1295  
dd15-p1305  
dd15-p1315  
dd15-p1325  
dd15-p1335  
dd15-p1345  
dd15-p1355  
dd15-p1365  
dd15-p1375  
dd15-p1385  
dd15-p1395  
dd15-p1405  
dd15-p1415  
dd15-p1425  
dd15-p1435  
dd15-p1445  
dd15-p1455  
dd15-p1465  
dd15-p1475  
dd15-p1485  
dd15-p1495  
dd15-p1505  
dd15-p1515  
dd15-p1525  
dd15-p1535  
dd15-p1545  
dd15-p1555  
dd15-p1565  
dd15-p1575  
dd15-p1585  
dd15-p1595  
dd15-p1605  
dd15-p1615  
dd15-p1625  
dd15-p1635

the Social Sciences & University of Koblenz-Landau Martin Becker University of Wurzburg wuerzburg.de Philipp Singer GESIS - Leibniz Institute for the Social Sciences & University of Koblenz-Landau Denis Helic Graz University of Technology Andreas Hotho University of Wurzburg and L3S Hannover wuerzburg.de Markus Strohmaier GESIS - Leibniz Institute for the Social Sciences & University of Koblenz-Landau ABSTRACT We present a new method for detecting interpretable subgroups with exceptional transition behavior in sequential data. Identifying such patterns has many potential applications, e.g., for studying human mobility or analyzing the behavior of internet users. To tackle this task, we employ exceptional model mining, which is a general approach for identifying interpretable data subsets that exhibit unusual interactions between a set of target attributes with respect to a certain model class. Although exceptional model mining provides a well-suited framework for our problem, previously investigated model classes cannot capture transition behavior. To that end, we introduce first-order Markov chains as a novel model class for exceptional model mining and present a new transition matrix measure that quantifies the exceptionality of transition subgroups. The measure compares the transition matrix of a subgroup and the respective matrix of the entire data with the transition matrix of random dataset samples. In addition, our method can be adapted to find subgroups that match or contradict given transition hypotheses. We demonstrate that our method is consistently able to recover subgroups with exceptional transition models from synthetic data and illustrate its potential in two application examples. Our work is relevant for researchers and practitioners interested in detecting exceptional transition behavior in sequential data. Keywords: Subgroup Discovery, Exceptional Model Mining, Markov chains, Transitions, Sequential Data 1. INTRODUCTION Exceptional Model Mining, a generalization of the classic subgroup discovery task, is a framework that identifies patterns which contain unusual interactions between multiple target attributes. In order to obtain operationalizable insights, it emphasizes the detection of easy-to-understand subgroups, i.e., it aims to find exceptional subgroups with descriptions that are directly interpretable by domain experts. In general, exceptional model mining-Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from KDD '16, August 13 - 17, 2016, San Francisco, CA, USA 2016 Copyright held by the owner / author. Publication rights licensed to ACM. ISBN 978-1-4503-4232-2/16/08...\$15.00 DOI: ing operates as follows: A target model of a given model class is computed once over the entire dataset, resulting in a set of model parameters. The same parameters are also calculated for each subgroup in a large (often implicitly specified) candidate set, using only the instances covered by the respective subgroup. A subgroup is considered as exceptional or interesting if its parameter values differ significantly from the ones of the overall dataset. While exceptional model mining has been implemented for a variety of model classes including classification, regression, Bayesian network and rank models, it has not yet been applied using models for sequential data. In this paper, we aim to apply exceptional model mining to discover interpretable subgroups with exceptional transition behavior. This enables a new analysis method for a variety of applications. As one example, assume a human mobility dataset featuring user transitions between locations. The overall transition model could for example show that people either move within their direct neighborhood or along main roads. Detecting subgroups with exceptional transition behavior goes beyond this simple analysis: It allows to automatically identify subgroups of people (such as "male tourists from France") or subsegments of time (such as "10 to 11 p.m.") that exhibit unusual movement characteristics, e.g., tourists moving between points of interest or people walking along well-lit streets at night. Other application examples could include subgroups of web-users with unusual transition behavior or subgroups of companies with unusual development over time. The main contribution of this paper is a new method that enables mining subgroups with exceptional transition behavior by introducing first-order Markov chains as a novel model class for exceptional model mining. Markov chains have been utilized for studying human mobility and analyzing human transition behavior, or to apply exceptional model mining with this model, we derive an transition matrix measure that quantifies the exceptionality of a subgroup's transition model. It measures how much the transition matrix of a subgroup and the respective matrix of the entire data deviates from the transition matrix of random dataset samples. This measure can be integrated into any known search algorithm. We also show how an adaptation of our approach allows to find subgroups specifically matching (or contradicting) given hypotheses about transition behavior (cf.). This enables the use of exceptional model mining for a new type of studies, i.e., the detailed analysis of such hypotheses. We demonstrate the potential of the proposed approach with synthetic as well as real-world data. 965 The remainder of this work is organized as following: We summarize our background in Section 2. Then, the main approach for mining subgroups with exceptional transition behavior is introduced in Section 3. Section 4 presents experiments and results. Finally, we discuss related work in Section 5, before we conclude in Section 6. 2. BACKGROUND Our solution extends Exceptional Model Mining with first-order Markov Chain Models. In the following, we give a brief overview of both techniques. 2.1 Exceptional Model Mining We formally define a dataset D as a multiset of data instances I described by a set of attributes A consisting of describing attributes AD A and model attributes AM A. A subgroup consists of a subgroup description p: D -> that is given by a Boolean function, and a subgroup cover c, i.e., the set of instances described by p, i.e., c = {I in I | p(I) = true}. In principle, our approach works with any pattern description language to describe

ignore  
11 ca  
12 kdd  
13 acm  
14 august  
15 copyright  
16 keywords  
17 san francisco  
18 introduction  
19 isbn  
20 experiments  
Method  
21 barabasi-albert  
22 progressive sampling  
23 heuristic  
24 monte-carlo, montecarlo  
25 non-linear optimization  
26 map-reduce, mapreduce  
27 en algorithm  
28 cmpp  
29 random sampling  
30 linear system, linear model  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47 large-scale network  
48 minimum-cut  
49 motif discovery  
50 rule discovery  
51 future event  
52 co-authorship network  
53 modularity  
54 paper recommendation  
55 eco-centric circle  
56 biological network  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000

## (2) Evaluating clustering analysis (K = 3) based on two PCA features.

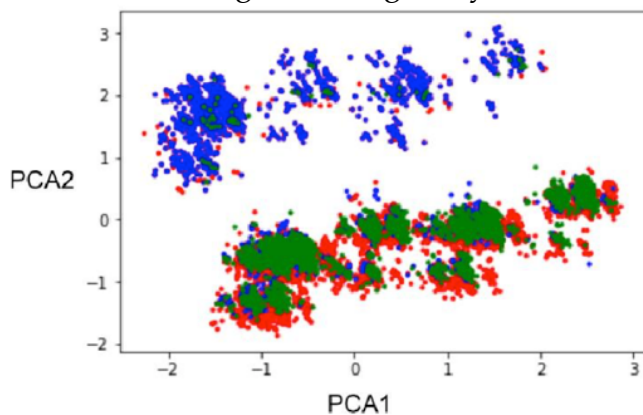


Fig.6 True label

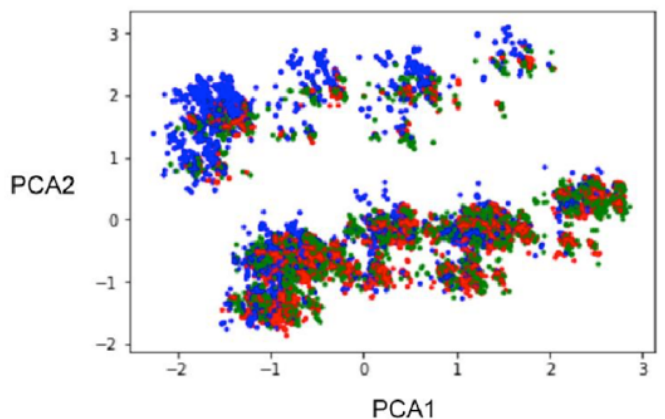
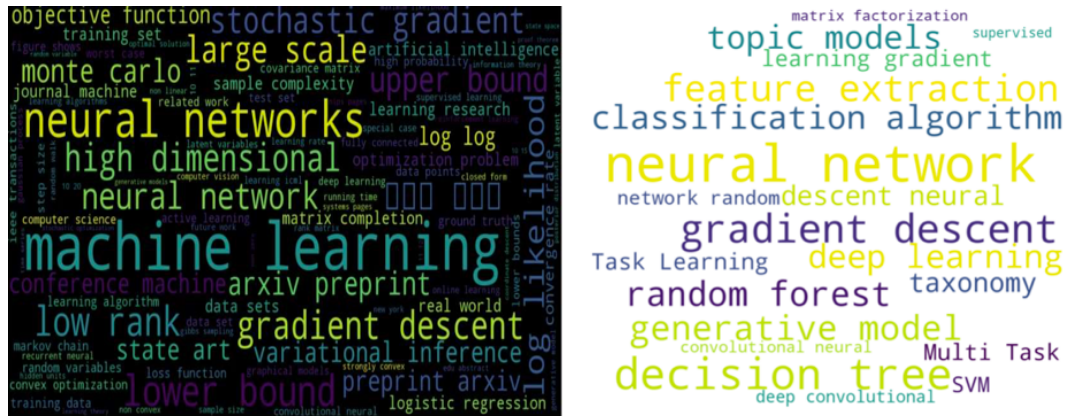


Fig.7 Clustering labels

### (3) Word cloud of entity clusters



(4) Classification performance (F1 score) vs training size (%) and #PCA features.

