Trust based recommendation engine

Team NASA – Technical Report

NASA Research Team:

Neeraj Saini
Venkatesh Sriram
Shuai Wang
Pujita Rao

Advisor: Dr. Jia Zhang 12/6/13 Graduate Practicum

TABLE OF CONTENTS

Project Summary

The project involved building a recommendation engine on top of the human-trust software which was the research work done by last year's NASA practicum Team. The engine will provide recommendations of existing reusable artifacts, workflows and models based on scientist's domain relevance and social connections. We used five features to calculate the likelihood of two authors to co-author in the future. We used machine learning algorithms such as Naïve based and JACARD similarity to work on the training data we extracted from DBLP (Which is essentially a database for all computer science publications). It also contains lot of material for earth science publications which we used as our data source. We provided our clients a proof of concept model which can easily be integrated with NASA tool vistrails and can also be run as a standalone service.

II. PROBLEM DOMAIN

Given two authors, find the probability that these authors will collaborate with each other (help each other). The major factor in recommendation of a tool is if the creator of the tool is someone who is likely to help you

III. PROJECT GOALS ACCOMPLISHED

Over the past few years, software development has seen a new paradigm where number of developers across the web, collaborate with each other in order to solve a big issue. This is accomplished by the concept of Application Programming Interfaces (API), where in a developer can reuse the work of another developer by using the APIs exposed.

This concept has become so prominent today, that we have a large number of software modules available to accomplish the same task, which are similar in terms of functional and non-functional aspects. And so the big question that comes up is - “How to find the best suited software modules to fulfill a task under a given context?”

In our project, we addressed this problem by building a prototype for an automated context aware recommendation framework which would leverage the social network data about the developer and draw out trust relationships in order to rate a software with respect to others.

IV. DATASET

- The DBLP Computer Science Bibliography (http://dblp.uni-trier/xml/)
- 2.3 Million publications dating back to 1936. Consistently updated since the 1980's
- Considered the definitive publication bibliography in Computer Science
- Readily available and complete dataset.
- Useful for building proof of concept before using actual Earth Science data
- Also contains a large number of Earth Science articles

DBLP

Category	Derivable Information	Description/Explanation
Publication	Publication Type	Includes: Article, Inproceeding, Proceeding, Book, Incollection, Ph.D Thesis, Master Thesis, and WWW. Each publication type possesses a different weight for different knowledge domains.
	# of Authors	In some contexts co-authorship ordering of names matters (e.g. in computer science the first listed name is regarded as the greatest contributor).
	# of Editors	Importance not yet established.
	# of Citations	The more citations used, the greater perceived knowledge for the publication. Additionally, recognizing that certain authors are repeatedly cited implies some social trust in that user.
	# of Times Cited	Being cited often implies high knowledge in domain. Being cited repeatedly by certain authors implies high social trust in author.
	Organization/School/Publisher Relationship	Domain specific, certain relationships present more value for both knowledge and social trust (e.g. Carnegie Mellon University increases domain score for computer science).
	Date Published	The older the paper, the less knowledgeable it is ranked (reason being that the content usually becomes dated. Exceptions exist, but they're few in number by comparison to all publications).
	Knowledge Domain(s)	Determined primarily through text mining the publication's title and keywords. Secondary would be abstract and citations. Tertiary would be the content of the content (introduction to conclusion). Quarternary would be to evaluate the knowledge domain(s) of the works citing the publication (potential inference of knowledge domains).
User	# of Publications Authored	The more publications an author produces, the greater their knowledge is perceived to be. Co-Authorship count also raises

Category	Derivable Information	Description/Explanation
		perceived social trust.
	# of Publications Edited	The more publications edited by a person, the greater their knowledge is perceived to be. Also raises perceived social trust, as responsibility is assigned to treat works authored by others. As a result, knowledge score is typically perceived greater than authorship, given vast domain related knowledge needed to be an editor.
	# of Publication Types Authored	Includes: Article, Inproceeding, Proceeding, Book, Incollection, Ph.D Thesis, Master Thesis, and WWW. The more publication channels/types authored by a person, the greater their breadth of knowledge is perceived to be (reason being that they are active in conferences, books, research fields, and more: must be highly competent to do this).
	# of Publication Types Edited	Includes: Article, Inproceeding, Proceeding, Book, Incollection, Ph.D Thesis, Master Thesis, and WWW. Same as for "# of Publication Types Authored", but additionally includes social trust to be charged with others' publications (as well as a boost to knowledge domain [for most presumed domains]; reason being that they are highly familiar with the particular knowledge domain across most publication channels).
	# of Times Cited by Others	The more often an individual is cited by others, the greater that person's knowledge is perceived to be. Social trust may also be perceived if repeatedly cited by certain authors across multiple publications (multiple publication types presumed to be insignificant here; warrants testing).
	# of Times Cited per Paper	The higher the averaged number, the greater the perceived knowledge of the individual cited. The reasoning for having an average score is to offset the impact of a statistical outliers/tails throwing off knowledge scores when taking into account "# of Times Cited by Others". For example, if someone has 100 publications, with 99 publications never being cited and 1 publication being cited over 9,000 times.
	# of Publications Produced in a Given Time Interval	This is a special case where the reviewer is interested in assessing knowledge of the reviewed individual during a specific time interval.
	# of Publications Cited Per Paper, Given a Time Interval	This is a special case where the reviewer is interested in assessing knowledge of the reviewed individual during a specific time interval.
	# of Publications Related to an Organization/School/Publisher	The more publications an individual has associated with a particular organization, school, or publisher – the higher their perceived knowledge and (marginally increased) social trust (due to continued interaction with people [potentially different people] associated with the entity).
	Date of Oldest Publication	The older the first publication, the greater the presumed knowledge of the individual (reason being that they've been involved in said knowledge domain for a long time).

Category	Derivable Information	Description/Explanation
	Date of Most Recent Publication	The more recent an individual's latest work is published, the higher their perceived knowledge (reason being that they're still actively contributing to the knowledge domain).
	Average Time Between Publishing	Lower scores may be perceived as a highly active individual in a particular domain; thus raising the perception of knowledge for the reviewed individual (reason being that we know the individual is actively engaged in the domain). This may be further extended for review within specific time intervals, at the reviewer's discretion.
	Percentage of Publications That are Domain Specific	Higher percentages raise the perception of knowledge for the reviewed individual (reason being that low percentages may be indicative of an individual who does not focus on a particular domain, but rather multiple domains. Basically, it's for evaluating a jack-of-all-trades publisher to a specialist publisher – the specialist is presumed to be more knowledgeable and would likely score a higher percentage).

- Elements:
  1. Article – an article from a journal or magazine.
  2. Inproceedings – An article in a conference proceedings.
  3. Proceedings – The proceedings of a conference (typically academic).
  4. Book – A book with an explicit publisher.
  5. Incollection – A part of a book having its own title. Basically, a publication within a book.
  6. Pdthesis – A Ph.D thesis.
  7. Masterthesis – A Master's thesis.
  8. www – Author homepage links (irrelevant?).

Entities:

a. Note: Definitions mostly taken from BibTeX (*http://en.wikipedia.org/wiki/BibTeX*)
b. Note: ~~Crossed out~~ entities indicate their not possessing much value for trust computing.

1. Author: The person(s) who write the text.
2. Editor: The person(s) who correct written works.
3. Title: Title for the written text.
4. Booktitle: Title of the book, if only part of it is being cited.
5. Year: The year of publication (if unpublished, then month of creation).
6. Journal: The journal or magazine the work is published in.
7. Month: Month of publication (if unpublished, then month of creation).
8. URL: Web Address.
9. Cite: Citations used in publication.
10. Publisher: At its core, the company that disseminates the written work.
11. School: The school where the thesis was written.
12. Crossref: The key of the cross-referenced entry.

13. Pages: Page numbers.
14. Address: Publisher's address (usually just city, but can be full address for lesser-known publishers).
15. Volume: The volume of a journal or multi-volume book.
16. Number: The "issue number" of a journal, magazine, or tech-report.
17. ee: "encyclopedia entry" or "electronic edition". In use, it's typically a web link to the author's text.
18. Cdrom: CD that publication is available on.
19. Note: Miscellaneous extra information.
20. Isbn: Special # often associated with written texts, particularly books. Used for identification/look-up purposes.
21. Citationcount: Number of citations present. Doesn't seem to work for everything. :/
22. Series: The series of books the book was published in.
23. Chapter: The chapter number.
24. Url: Web link to paper.

- Limitations:
  1. Doesn't take into account (entities): Conference papers, booklets, manuals, tech reports (school-based), published documents, and other miscellaneous written texts.
  2. DBLP lacks a description for elements and entities. The best approach at defining items is to manually review the xml files, review prior research by others on DBLP, and take into account that DBLP is basically a degraded version of BiBtEx + XML. Even the dtd file (http://dblp.uni-trier.de/xml/dblp.dtd) doesn't contain all of the entities used within the XML files. Very misleading.

The Below Table: Shows what entities are associated with each element.

Element Type	Entity Type (R = Required; O = Optional)
Article	R: Author
	R: Title
	R: Journal
	R: Year
	O: Volume
	O: Number
	O: Pages
	O: Month
Inproceeding	R: Author
	R: Title
	R: Book Title
	R: Year
	O: Editor

Element Type	Entity Type
	O: Volume/Number
	O: Series
	O: Pages
	O: Address
	O: Month
	O: Organization
	O: Publisher
Proceeding	R: Title
	R: Year
	O: Editor
	O: Volume/Number
	O: Series
	O: Address
	O: Month
	O: Publisher
	O: Organization
Incollection	R: Author
	R: Title
	R: Book Title
	R: Publisher
	R: Year
	O: Editor
	O: Volume/Number
	O: Series
	O: Type
	O: Chapter
	O: Pages
	O: Address
	O: Edition
	O: Month
Book	*R: Author
	*R: Editor
	R: Title
	R: Publisher
	R: Year
	O: Volume/Number
	O: Series
	O: Address
	O: Edition
	O: Month
Ph.D Thesis	R: Author
	R: Title
	R: School
	R: Year

Element Type	Entity Type
	O: Type
	O: Address
	O: Month
Master Thesis	R: Author
	R: Title
	R: School
	R: Year
	O: Type
	O: Address
	O: Month
WWW	R: URL

*Table Note: Either an “editor” *or* an “author” is required for a book entry.

ANALYSIS OF COMPLETE DATASET

Nature of dataset : XML file of size 1.2 GB.

Parsing challenges:

- Estimated Time taken
  - Time taken to create Co-Author Graph (explained later) on full DBLP file ≈ 22 minutes (using a Macbook Pro with 8GB RAM and 2.6 GHz core i5)
  - Estimated number of such calculations required for complete dataset ≈ 6 million
  - Thus, estimated time taken on our machine≈ 250 years!

Additional challenges:

Possibility of Overfitting

- A learning algorithm is said to Overfit, if it is more accurate in fitting known data (hindsight) but less accurate in predicting new data (foresight).
- Usually occurs due to rare examples that may have no causal relation to target function.
- Some author pairs are very unlikely to coauthor (due to different research interests or different time frames). The data calculation for these author pairs could cause Overfitting.

STRATEGY

- Focus on one particular research area (such as distributed systems)
- Smaller but complete dataset (for the domain)
- Took all publications related to Distributed Systems (1900 publications and 1082 distinct authors)

Advantage: We can expect convincing and representative results due to completeness of dataset within the matched domain.

GENERATING TRAINING DATA

Modeling the problem-

Given the collaboration history of a huge set of authors, find out the factors that affect their willingness to collaborate, and fit those factors in a machine learning model.

NAÏVE BAYES network

- Classification algorithm for calculating the probability of an outcome.
- Suitable for problems where a small set of features affect final outcome, and these features are independent of each other.
- Classifies the output as one out of a small set of possibilities.
- Formal definition-

The probability model is a conditional model

C – Output with small set of possibilities

$F_1 \ldots F_n$ *– Independent input features*

The likelihood of co-authorship can be modeled as a Naïve Bayes classification problem.

$$p(C| F_1, F_2, F_3, F_4, F_5)$$

Output (C) - Probability that two authors will co-author.

Independent input features-

F_1- Similarity of research areas

F_2- Similarity of author reputation

F_3- Author connectedness

F_4- Collaboration history of each author

F_5- Work interaction strength

JACCARD SIMILARITY

- Statistic used for comparing the similarity and diversity of input sets.
- Given two sets of inputs, it outputs a percentage similarity between the two sets.
- Formal definition-

Let A and B be the input sets, J(A,B) is the Jaccard similarity of the two sets.

F1 - SIMILARITY OF RESEARCH AREAS

Rationale- Two authors are more likely to co-author when they share the same research areas.

Modeling the problem-

Given two authors and their publication information, calculate the similarity of their past areas of research interest

F2 - SIMILARITY OF AUTHOR REPUTATION

Rationale- Two people are more likely to co-author with each other if their reputation in the field is similar i.e. a person with high reputation is more likely to collaborate with another person with high reputation, and vice versa.

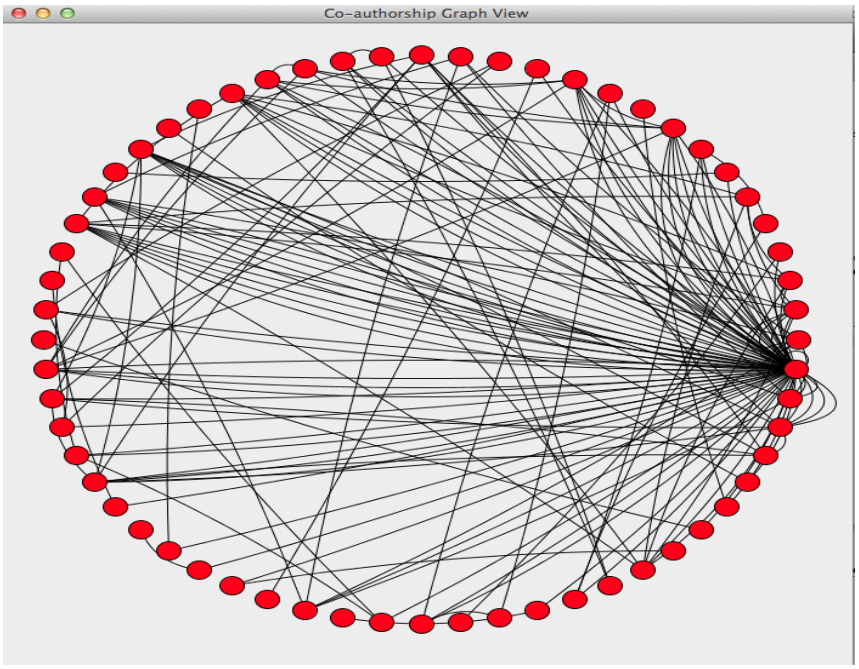
Assumption- We consider the number of cases of a professor co-authoring with a student as sufficiently small.

CO-AUTHOR GRAPH

Formal definition-

We can represent Co-authorship as a Graph G = (V , E), such that

- V is Vertex Set, each vertex is an Author
- E is Edge set, where an Edge exists between two authors, if they have Co-Authored on anything in the past.



Explanation-

Let a sub-graph of the Co-Author graph be represented by the following

Authors– A,B,C,D,E,F

Past Co-Authorship exists between-

A and B,

A and E,

B and E,

C and D,

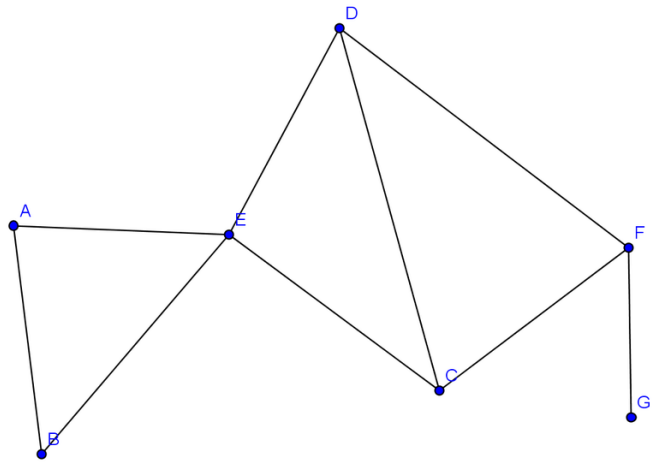
C and E,

C and F,

D and E,

D and F,

F and G



F3 - AUTHOR CONNECTEDNESS

Rationale- Two authors are more likely to collaborate with each other if they have collaborated in the past, or if they have a common author with whom they have both collaborated in the past.

Assumption:-

In the co-author graph,

Co-authorship set (CS)

- Definition- Given an author A, Co-Authorship set of A is represented as CS(A). It is the set of all the people that the author has ever co-authored with.
- It is the number of Nodes adjacent to Node A, in the Co-Author Graph

Example-

Here, CS(A) is (B,C,D)

F4 –COAUTHORSHIP INCLINATION

Rationale-

- An author who has collaborated with a large number of people in the past, is more likely to collaborate with new people.

Assumptions-

- Co-Authorship inclination of an author A is indicated by the Size of Co-Authorship Set of A.
- Both authors need to have a good Co-Authorship Inclination in order to collaborate

AFFILIATION SIMILARITY

Rationale-

- Two authors are more likely to collaborate if they have some common affiliation
- For example, two people who are part of the same university would be more likely to collaborate with each other.
- NOT implemented yet

Analysis of findings

Purpose – To fine-tune machine learning model based on experimental results.

Metrics used-

1) Precision = $\frac{|Cr|}{|C|}$

2) Recall = $\frac{|Cr|}{|R|}$

C – Set of author pairs categorized as ”Y” i.e. likely to co-author. |C|- Size of C.

R - Set of author pairs that actually co-authored

|R|- Size of R.

C_R - Set of author pairs correctly classified

$|C_R|$- Size of C_R

Calculations –

1) Recall :

Number of records used as test set = 725

Number of actual co-author pairs = 130

Number of co-author pairs correctly identified = 126

Recall value = $\frac{126}{130}$ = 0.9718

i.e. Recall percentage = 97.18%

2) Precision :

We used publications published after 2008 to calculate precision

Number of records = 351

Number of co-author pairs predicted = 109

Number of actual co-author pairs = 25

Precision value = 1- ((109−25)/351) = 0.7606

Precision percentage = 76.06%

DEPLOYMENT INSTRUCTIONS

GETTING RESULTS IN REST API USING JAVA PLAY!

1) First, you need to install Java Play!. This is a very simple process, and you just need to do what is said in http://www.playframework.com/documentation/2.0/Installing
2) In the terminal, navigate to the Project source folder. Then type the following-

   play run

   This sets up the server and your play app is ready to go.

3) The following are the URL's you need to type in your browser
NOTE - Unless you host it, it should be after localhost:9000 (e.g. localhost:9000/getReputation/Otto Muzik)

a) Get a graph of the co-authors of an author given a level and a name, returns in JSON format

/getGraphWithoutRender/:name/:level

b) Same as getGraphWithoutRender except that it also calls the JUNG framework to render the graph, if you have it installed
/getGraphWithRender/:name/:level

c) Given an author's name, get all the co-authors, JSON Format
/getCoAuthorInformation/:name

d) Given an author's name (part of the dataset), it returns the trust value of the author
/getReputation/:name

e) Given an author's name (part of the dataset), and a topic, it returns the trust value of the author for that topic
/getReputationForTopic/:name/:topic

f) Given an author's name (part of the dataset), it returns the entire co-author graph beginning with that author
/getSocialNetwork/:name

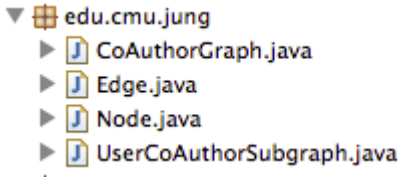
g) Given an author's name (part of the dataset), and a topic, it returns all the co-authors for that topic
/getCoAuthorsByTopic/:name/:topics

h) Given an author's name (part of the dataset), a topic, and a year, it returns all the co-authors for that topic after that year
/getCoAuthorsByTopicAndTime/:name/:topics/:year

4) In case you want to host it on Heroku, the following page will help
https://blog.heroku.com/archives/2011/8/29/play

VISUALIZING CO-AUTHOR GRAPHS

The package called *edu.cmu.jung* is the package that contains all the files related to creating and visualizing the co-author graph.

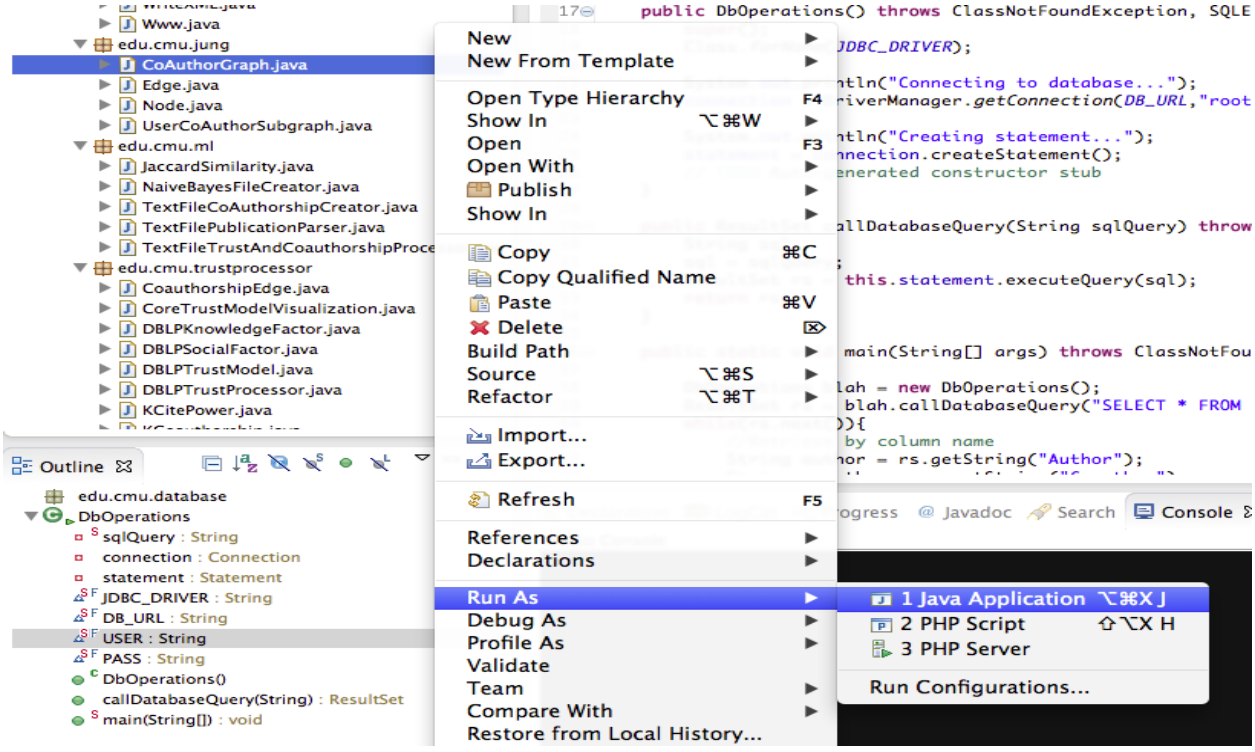


To visualize the entire co-author graph

Ensure that the project folder contains the XML file based on which you want to generate the co-author graph.

The steps you need to take are-

1) Go to *edu.cmu.jung*
2) Run the CoAuthorGraph.java file

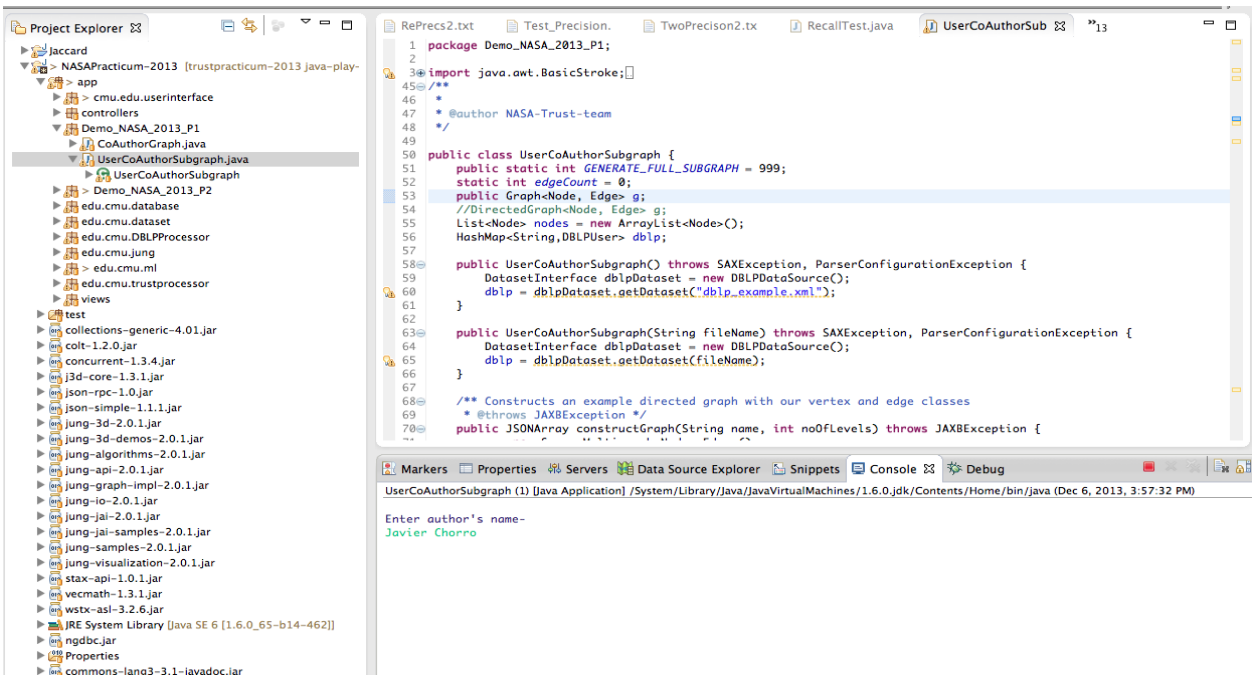


3) You get the output as both a graph, and a JSON representation in the console. This graph is based on *dblp_example.xml*

The steps you need to take are-

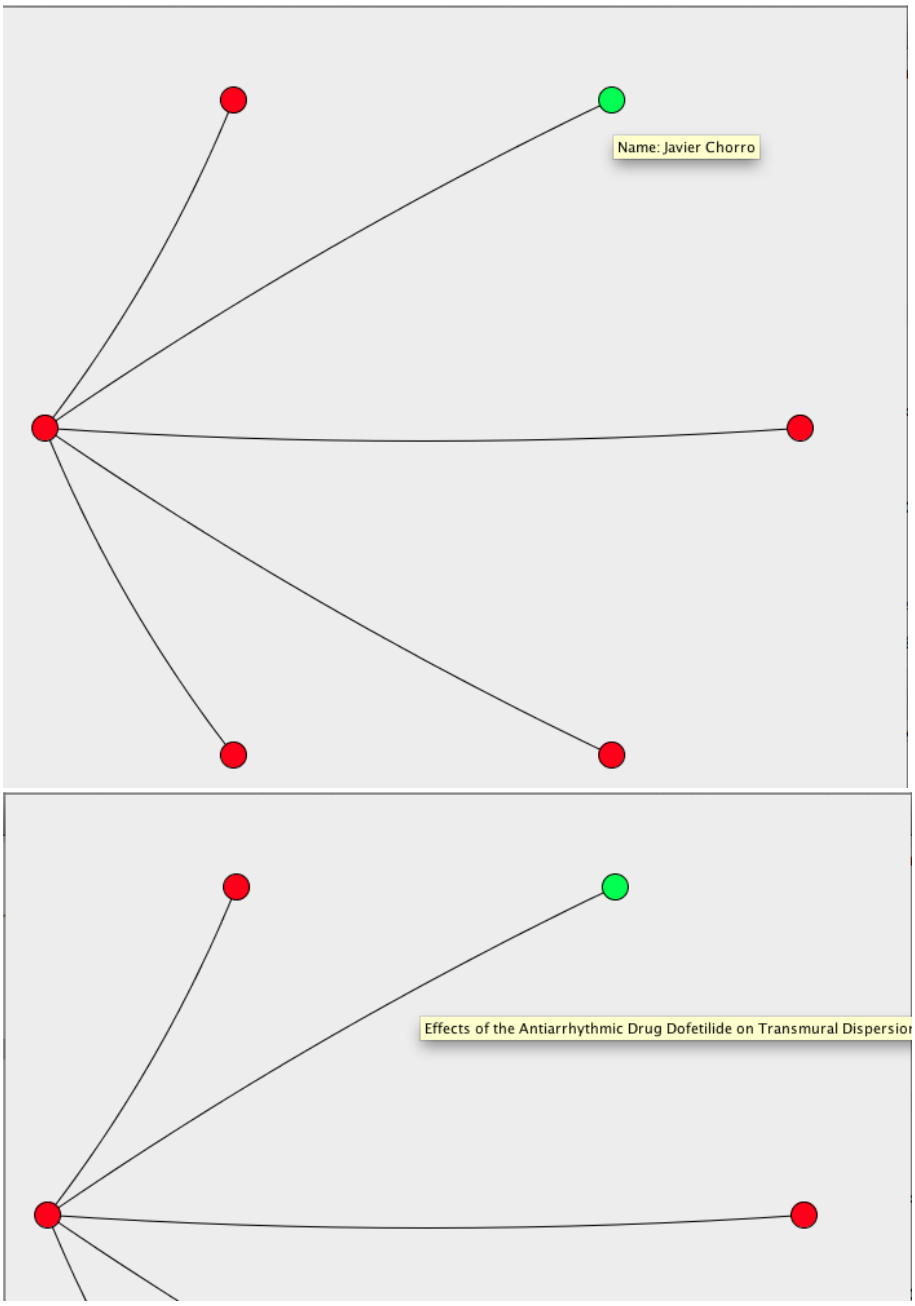
1)Go to *Demo_NASA_2013_P1*

2)Run the UserCoAuthorSubGraph.java file, in the console put the author name you are interseted.



3) You get the output as both a graph, and a JSON representation in the console. This graph is based on *dblp_example.xml.*

Output screen shots:



4) If you hover over your mouse to one of the nodes it shows the author name and the edge will show the list of their publications.

MACHINE LEARNING AND DATABASE MANAGEMENT

1) CSV FILE PREPARATION-

1) Put dblp.xml in your project directory.
2) Run splitfiles.py on dblp.xml python splitfiles.py dblp.xml

This splits dblp.xml into 100 files named <split_dblp_40.xml> which makes it manageable to parse using JAXB.

3) Go to edu.cmu.DBLPProcessor package
4) Run FullDBLPDataToTextConverter.java

```
public static void main(String args[]) throws JAXBExceptior
    for(int i = 81; i<=100 ; i++) {
        int mb = 1024*1024;
```

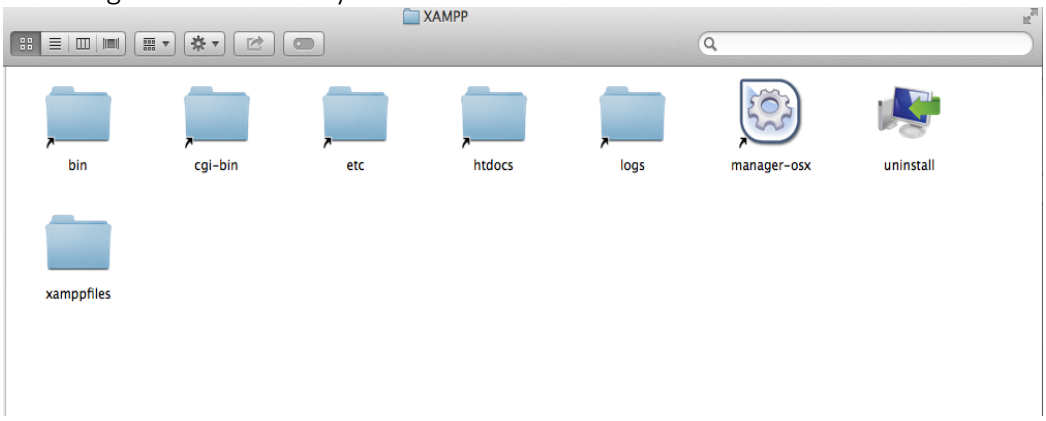
Here, i represents the file range on which to conduct the operations (i.e. split_dblp_i.xml).

5) Run the program 5 times, each time changing the range of i in sets of 20 (i.e. 1-20, 21-40… 81-100). This is because the Java Heap runs out if you do all 100 at the same time.
6) This puts in all the data in a file called dblpdata.txt.
7) To get the lines relevant to the topics that you want, make the required changes to *getrelevantlines.py* and run that python script on *dblpdata.txt.*

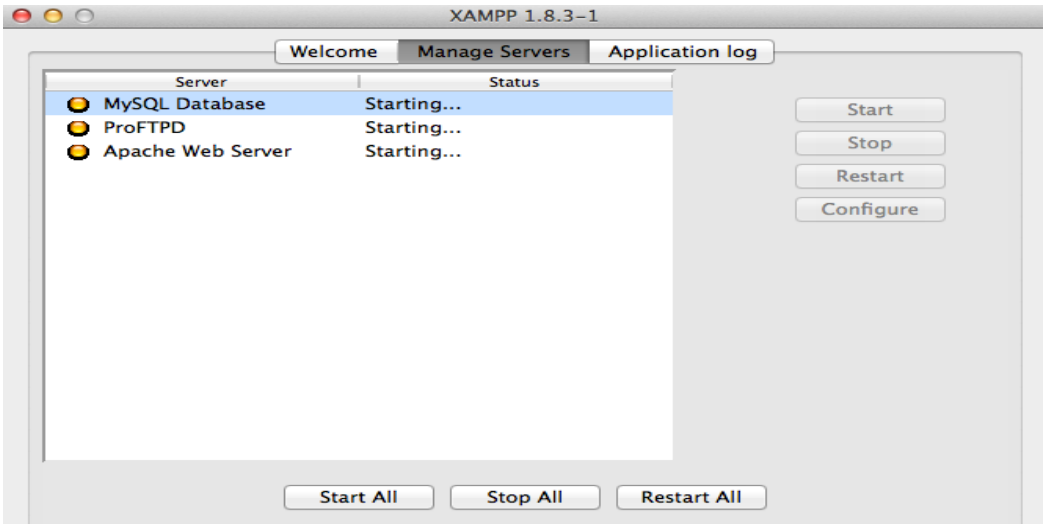
NOTE:- YOU SHOULD NOT NEED TO DO THIS. THE RELEVANT CSV FILE IS IN dblp.csv IN PROJECT DIRECTORY

2) DATA PREPARATION

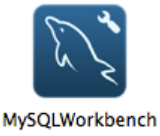
- To get the DBLP and retrieve data dynamically, we are using mysql database. Download link: http://dev.mysql.com/downloads/
- Also click on build configuration to add mysql-connector-java-5.1.27-bin.jar file. It is under the folder after you download mysql
- Also download MySql workbench for database management, each time you want to run the code, make sure it is up running, I am using XAMPP library for mac, link:http://sourceforge.net/projects/xampp/files/XAMPP%20Mac%20OS%20X/ the following shows the library content:



- Click on manager—osx, and click on Mysql database will make sure that it is started. Also Mysql Workbench can allow you to check database status too.



- When you have downloaded everything, the icon looks like:



- Open the workbench, we need to construct the database schema first, then two database tables will be used later, including Coauthors and Publications
  Table column screen shot:

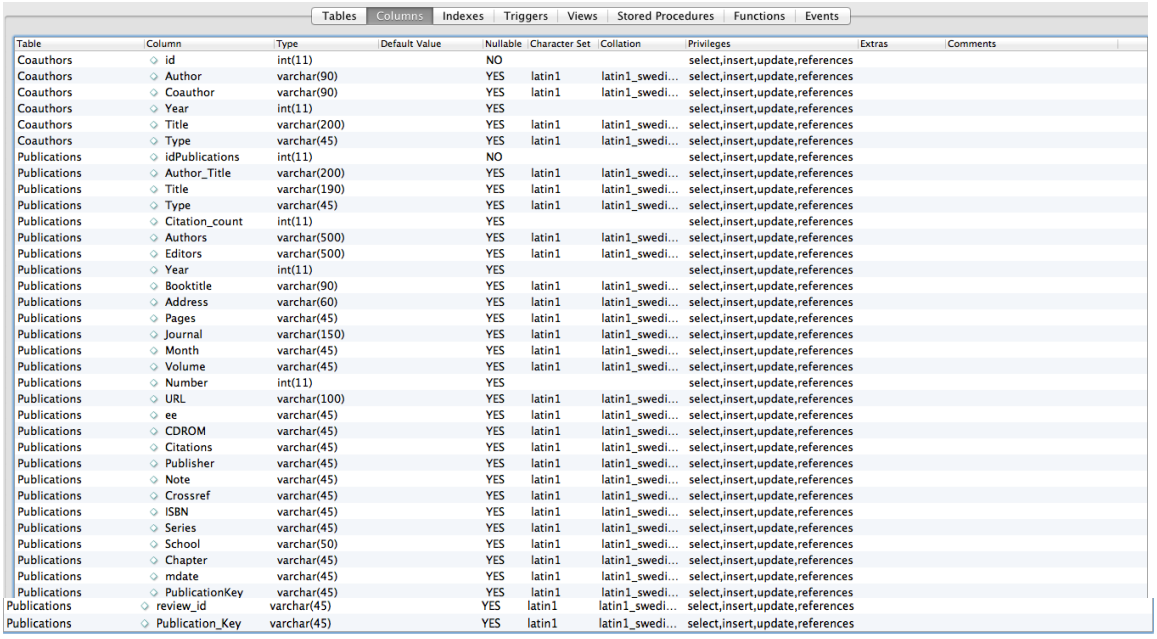


Table	Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extras	Comments
Coauthors	id	int(11)		NO			select,insert,update,references		
Coauthors	Author	varchar(90)		YES	latin1	latin1_swedi...	select,insert,update,references		
Coauthors	Coauthor	varchar(90)		YES	latin1	latin1_swedi...	select,insert,update,references		
Coauthors	Year	int(11)		YES			select,insert,update,references		
Coauthors	Title	varchar(200)		YES	latin1	latin1_swedi...	select,insert,update,references		
Coauthors	Type	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	idPublications	int(11)		NO			select,insert,update,references		
Publications	Author_Title	varchar(200)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Title	varchar(190)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Type	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Citation_count	int(11)		YES			select,insert,update,references		
Publications	Authors	varchar(500)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Editors	varchar(500)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Year	int(11)		YES			select,insert,update,references		
Publications	Booktitle	varchar(90)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Address	varchar(60)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Pages	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Journal	varchar(150)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Month	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Volume	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Number	int(11)		YES			select,insert,update,references		
Publications	URL	varchar(100)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	ee	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	CDROM	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Citations	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Publisher	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Note	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Crossref	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	ISBN	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Series	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	School	varchar(50)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Chapter	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	mdate	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	PublicationKey	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	review_id	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		
Publications	Publication_Key	varchar(45)		YES	latin1	latin1_swedi...	select,insert,update,references		

- Before importing CSV data to database we need to:
  Converting Full DBLP data to CSV

- After getting the csv files, we can click on Import button in the middle and import the data manually.

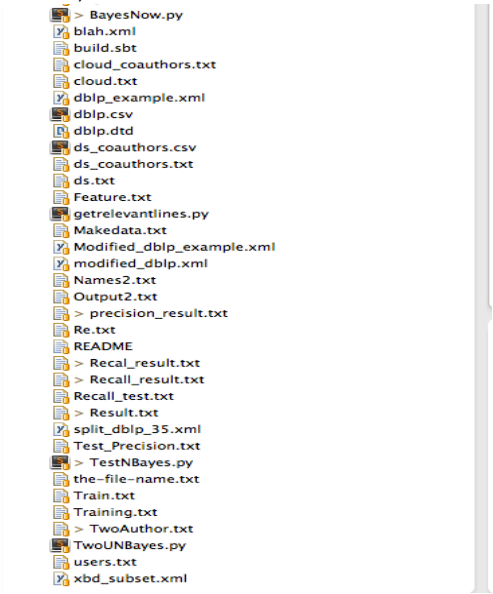
- Open the workbench, part of the data in table coauthors is shown as follows:

id	Author	Coauthor	Year	Title	Type
1	José Antonio...	Jose M. Giron...	2002	Design of a di...	article
2	Jose M. Giron...	José Antonio...	2002	Design of a di...	article
3	Fathi Tenzekhti	Khaled Day	2002	On fault-toler...	article
4	Fathi Tenzekhti	Mohamed Oul...	2002	On fault-toler...	article
5	Khaled Day	Fathi Tenzekhti	2002	On fault-toler...	article
6	Khaled Day	Mohamed Oul...	2002	On fault-toler...	article
7	Mohamed Oul...	Fathi Tenzekhti	2002	On fault-toler...	article
8	Mohamed Oul...	Khaled Day	2002	On fault-toler...	article
9	Ludovic Apvrille	Pierre de Saqu...	2006	TURTLE-P: a U...	article
10	Ludovic Apvrille	Ferhat Khendek	2006	TURTLE-P: a U...	article
11	Pierre de Saqu...	Ludovic Apvrille	2006	TURTLE-P: a U...	article
12	Pierre de Saqu...	Ferhat Khendek	2006	TURTLE-P: a U...	article
13	Ferhat Khendek	Ludovic Apvrille	2006	TURTLE-P: a U...	article
14	Ferhat Khendek	Pierre de Saqu...	2006	TURTLE-P: a U...	article
15	Ewa Deelman	Gurmeet Singh	2005	Pegasus: A fra...	article
16	Ewa Deelman	Mei-Hui Su	2005	Pegasus: A fra...	article
17	Ewa Deelman	James Blythe	2005	Pegasus: A fra...	article
18	Ewa Deelman	Yolanda Gil	2005	Pegasus: A fra...	article
19	Ewa Deelman	Carl Kesselman	2005	Pegasus: A fra...	article
20	Ewa Deelman	Gaurang Mehta	2005	Pegasus: A fra...	article
21	Ewa Deelman	Karan Vahi	2005	Pegasus: A fra...	article
22	Ewa Deelman	G. Bruce Berri...	2005	Pegasus: A fra...	article
23	Ewa Deelman	John Good	2005	Pegasus: A fra...	article
24	Ewa Deelman	Anastasia C. L...	2005	Pegasus: A fra...	article
25	Ewa Deelman	Joseph C. Jacob	2005	Pegasus: A fra...	article
26	Ewa Deelman	Daniel S. Katz	2005	Pegasus: A fra...	article
27	Gurmeet Singh	Ewa Deelman	2005	Pegasus: A fra...	article
28	Gurmeet Singh	Mei-Hui Su	2005	Pegasus: A fra...	article
29	Gurmeet Singh	James Blythe	2005	Pegasus: A fra...	article
30	Gurmeet Singh	Yolanda Gil	2005	Pegasus: A fra...	article
31	Gurmeet Singh	Carl Kesselman	2005	Pegasus: A fra...	article
32	Gurmeet Singh	Gaurang Mehta	2005	Pegasus: A fra...	article

- Part of the data in Publications table is shown as follows:

idPublications	Author_Title	Title	Type	Citation_count	Authors	Editors	Year	Booktitle	Address	Pages	Journal
1	José Antonio...	Design of a di...	article	0	José Antonio...		2002			207-213	Micropr
2	Jose M. Giron...	Design of a di...	article	0	José Antonio...		2002			207-213	Micropr
3	Lanfranco Lop...	Object and pr...	article	0	Lanfranco Lop...		2000			587-595	Micropr
4	Fathi Tenzekh...	On fault-toler...	article	0	Fathi Tenzekh...		2002			301-309	Micropr
5	Khaled Day_O...	On fault-toler...	article	0	Fathi Tenzekh...		2002			301-309	Micropr
6	Mohamed Oul...	On fault-toler...	article	0	Fathi Tenzekh...		2002			301-309	Micropr
7	Parveen Kuma...	A low-cost hy...	article	0	Parveen Kumar		2008			13-32	Mobile I
8	Vahid Garousi...	Incorporating...	article	0	Vahid Garousi		2010			113-137	Softwar
9	Ludovic Apvril...	TURTLE-P: a U...	article	0	Ludovic Apvril...		2006			449-466	Softwar
10	Pierre de Saqu...	TURTLE-P: a U...	article	0	Ludovic Apvril...		2006			449-466	Softwar

- After database configuration, we can open the project folder in eclipse:
  Make sure that BayesNow.py, TestNBayes and TwoUNBayes.py are under the project folder,

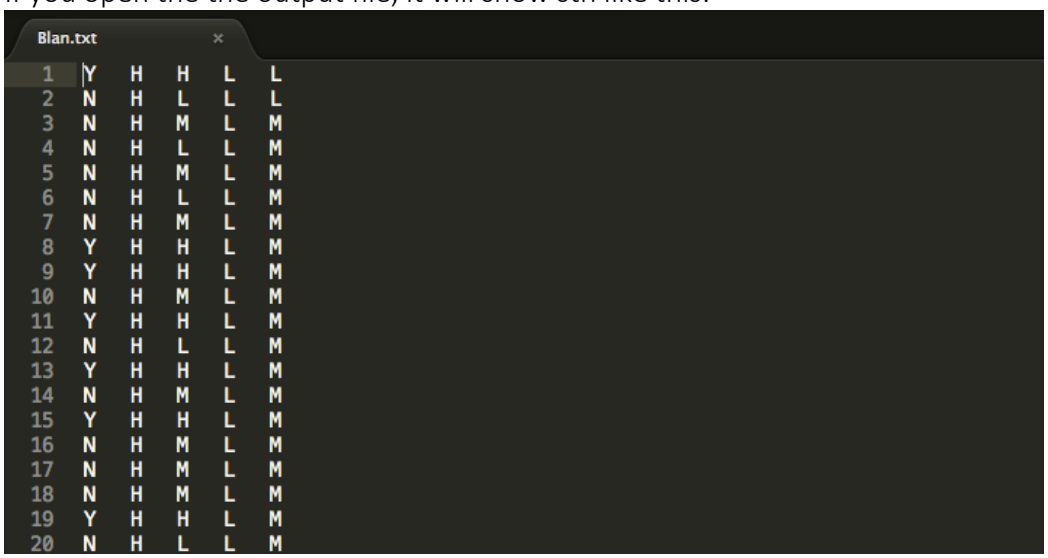


Generate training data and test data:

- Run the code NaiveBayesFileCreator under edu.cmu.ml package, it should print out sth like this, u can specify the file path in the main function second parameter.

```
<terminated> NaiveBayesFileCreator [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Dec 6, 2013, 3:13:19 PM)
Same elements [Replication techniques for speeding up parallel applications on distributed systems., Concurrency - Practice and Experience]
0.5
Same elements [Replication techniques for speeding up parallel applications on distributed systems., Concurrency - Practice and Experience]
0.5
Same elements [Replication techniques for speeding up parallel applications on distributed systems., Concurrency - Practice and Experience]
1.0
Same elements [Replication techniques for speeding up parallel applications on distributed systems., Concurrency - Practice and Experience]
0.5
Same elements [Replication techniques for speeding up parallel applications on distributed systems., Concurrency - Practice and Experience]
0.5
Same elements [Replication techniques for speeding up parallel applications on distributed systems., Concurrency - Practice and Experience]
1.0
Same elements [Replication techniques for speeding up parallel applications on distributed systems., Concurrency - Practice and Experience]
0.5
```

- If you open the the output file, it will show sth like this:



- Then you can randomly choose part of the records (10% recommendated) as test file, the other 90% as training file.

Generate feature file:

- Create a file called Feature.txt, the content should be like this:

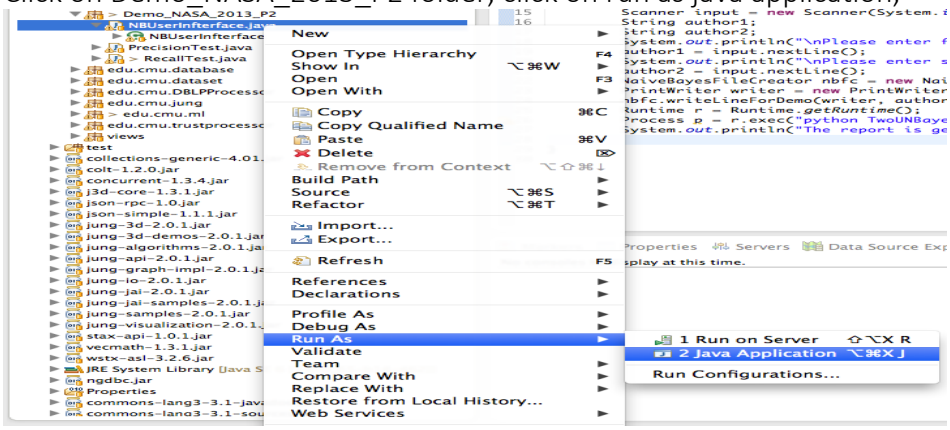
Feature_Name {Set_of_Values}
domain-similarity{ H, M, L}
reputation-similarity { H, M, L}
connectedness-similarity { H, M, L}
coauthorship-history{H,M,L}

Anytime you want to add more feature just add another line

3) Interfaces to use

Show probability of two specific authors to collaborate:

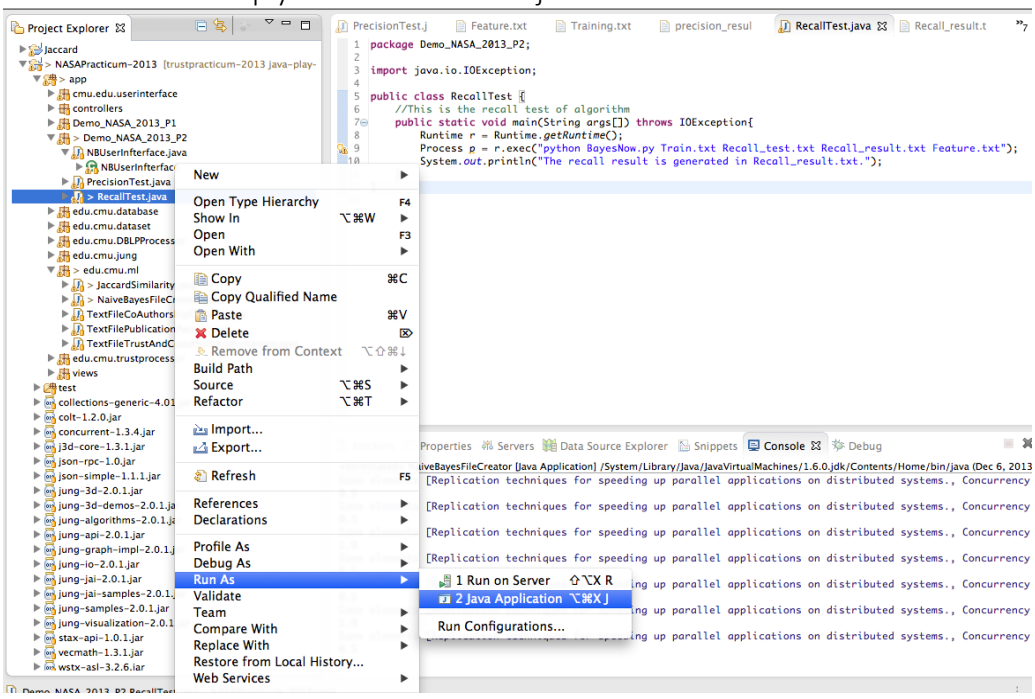
- Click on Demo_NASA_2013_P2 folder, click on run as java application,



Then you will input two authors name as input to see their collaboration probability and also the detailed the feature information.

Precision test and recall test:

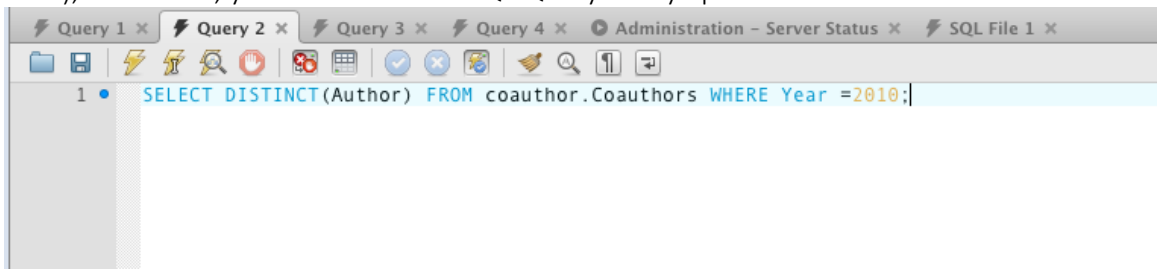
You can simply run the RecallTest.jave file like this:



Also u can use any data you created as input, the second parameter will be the training file and third one are the test files. Finally you can see a report generated in the project folder called Recall_result.txt. (Which you can easily rename as method input)

For precision test:

We need to have the real co-authorship information in real life (any co-authorship after 2010), to do that, you need to run a SQL Query in Mysql Workbench:



Then click on export data icon to one csv file. I am naming it Au2020.csv.
Later see the PrecisionTest.java line 29, put the name of your file there as input parameter like:

br=newBufferedReader(newFileReader("/Users/ShuaiWang/Desktop/Au2010.csv"));
Then you can run the code, it will actually take a while but the output should give you a number, in my experiment it shows: 26
Then you can refer to the precision_result.txt, it will show detail of each input, scroll down to the end of the file it shows:

```
2458 ----------------------------------
2459 How many we classified:109
2460 Total test number: 351
2461 Precision: 76.0683760684%
2462
```