Code Description

There are three parts of the code：Java、Matlab and Database。

# Java

Mainly including some preprocessor:

* separateXmlFiles is used to break up large xml file, so easy to follow-up processing。
* UsingAccessDatabase is used to read the xml file, and then organize the data and write it to the Access database。
* WordCount is used to count the frequency of words appearing in the title。

# Matlab

Mainly include computing of semantic vector, and the code of GUI interface design. The “cleanData” in the Database should be configured to the ODBC database, because the need to use ODBC way to call the database.

* getAuthorVector.m：Calculate the author's semantic vector
* getWordsVector.m：Calculate the semantic vector of words
* getIndex.m：Get the index
* normalization.m：Normalized Semantic Vector
* SearchForAuthor.m：Find keywords and authors of an author's related field
* AuthorSearch.m：GUI interface

# Database

It is the main part of the program, which is mainly on the database integration and code query. SQL language is used to query code.

* Authors table：All of the authors and their articles

code：

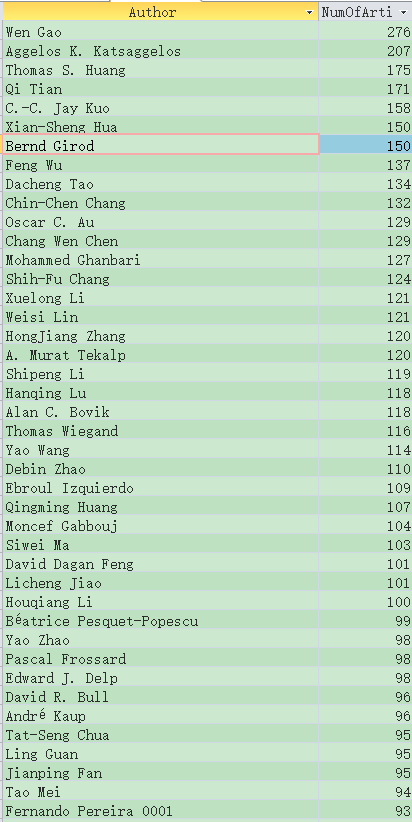
SELECT Article\_Author.[Author] AS Author, COUNT(\*) AS NumOfArticles

FROM Article\_Author

GROUP BY Author

ORDER BY COUNT(\*) DESC;

result：Total 130516 records



* Authors\_Frequent\_1（sup>2）table：The number of authors and their articles with a support count greater than 2 (a frequent set)

code：

SELECT Article\_Author.[Author] AS Author, COUNT(\*) AS NumOfArticles

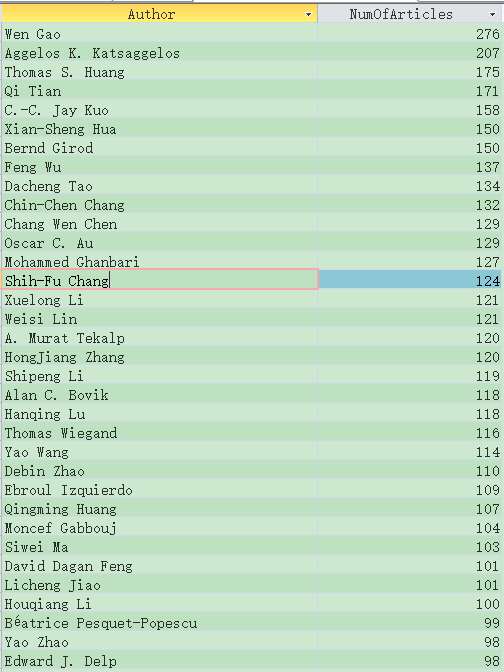
FROM Article\_Author

GROUP BY Author

HAVING (((Count(\*))>2))

ORDER BY COUNT(\*) DESC;

Result ：Total 25662 records



* Article\_Author\_Frequent\_1 table：connection between Authors\_Frequent\_1（sup>2）table and Article\_Author table

code：

SELECT Article\_Author.title AS title, Article\_Author.Author AS Author

FROM Article\_Author, [Authors\_Frequent\_1（sup>2）]

WHERE Article\_Author.Author = [Authors\_Frequent\_1（sup>2）].Author

ORDER BY title DESC;

result：Total 182351 records



* Authors\_Frequent\_2（sup>2）table：The number of the author 1、the author 2 and there cooperation articles with a support count greater than 2 (two frequent sets)

code：

SELECT Article\_Author\_Frequent\_1.[Author] AS Author1, Article\_Author\_Frequent\_1\_copy.[Author] AS Author2, COUNT(\*) AS NumOfCoArticles

FROM Article\_Author\_Frequent\_1,Article\_Author\_Frequent\_1 AS Article\_Author\_Frequent\_1\_copy

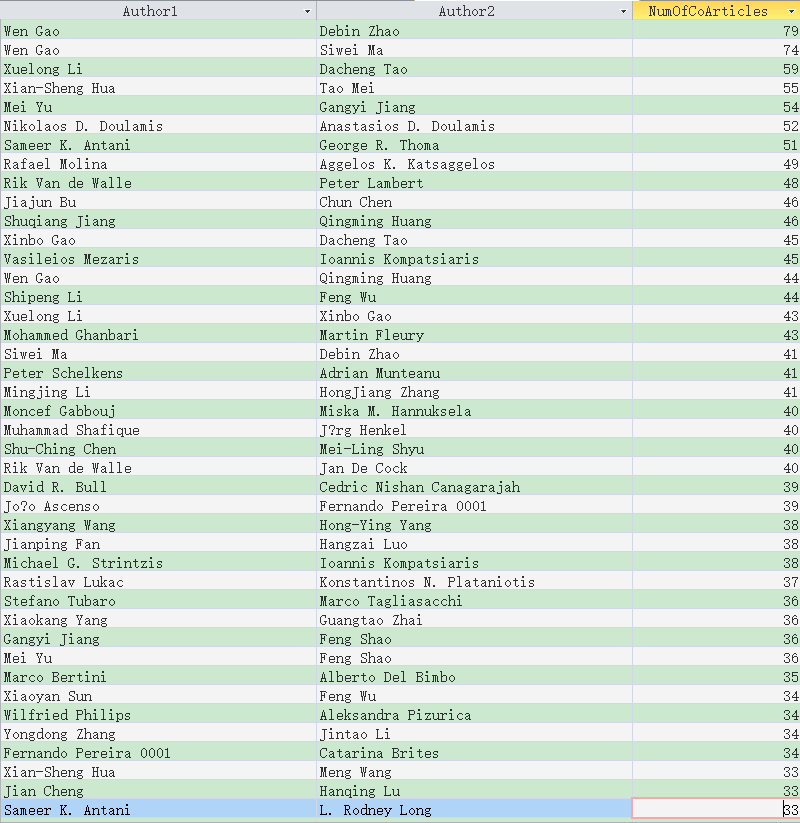
WHERE Article\_Author\_Frequent\_1.title=Article\_Author\_Frequent\_1\_copy.title AND Article\_Author\_Frequent\_1.[Author]>Article\_Author\_Frequent\_1\_copy.[Author]

GROUP BY Article\_Author\_Frequent\_1.[Author], Article\_Author\_Frequent\_1\_copy.[Author]

HAVING (((Count(\*))>2))

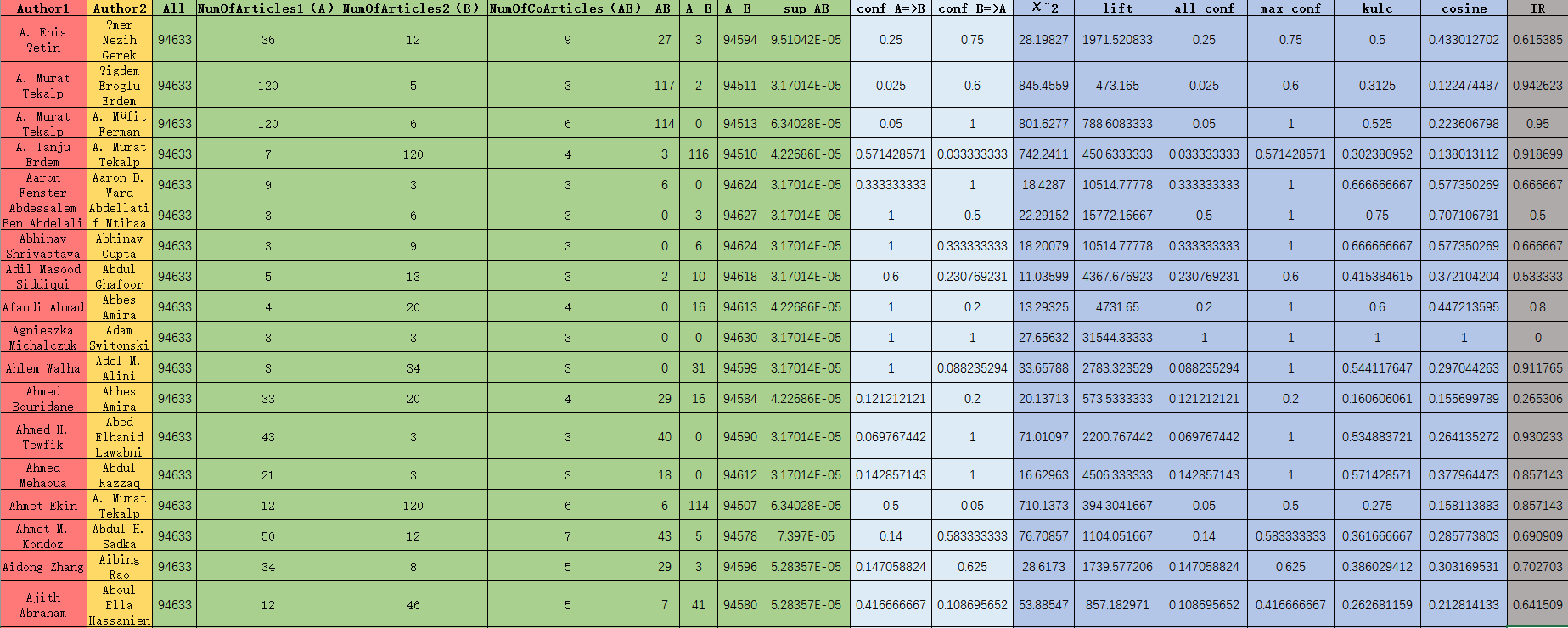
ORDER BY COUNT(\*) DESC;

result：Total 24633 records



* Evaluate the relationship between the two frequent sets with the six pattern assessment methods, and evaluate which relationship is better

Put the statistical results into the excel table, and calculate the various models of assessment methods according to the formula



* Assessment of teacher-student relationship

Need to find each author first article published time FirstPubYear

* Author table：

code：

SELECT Article\_Author.[Author] AS Author, COUNT(\*) AS NumOfArticles, MIN(Articles.year\_) AS FirstPubYear

INTO Authors

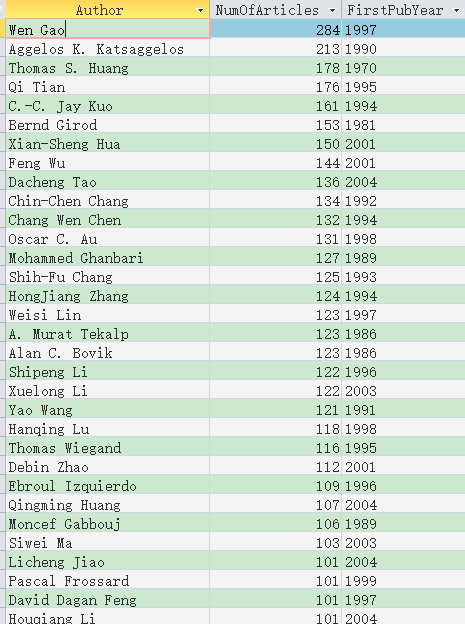
FROM Article\_Author, Articles

WHERE Articles.title=Article\_Author.title

GROUP BY Author

ORDER BY COUNT(\*) DESC;

result：total 130516 records



Need to find the number of coauthor of per author

* NumOfCoAuthors table：

code：

SELECT DISTINCT Article\_Author.Author AS Author, Article\_Author\_copy.Author AS CoAuthor

FROM Article\_Author, Article\_Author AS Article\_Author\_copy

WHERE Article\_Author.title = Article\_Author\_copy.title AND Article\_Author.Author<>Article\_Author\_copy.Author;结果：

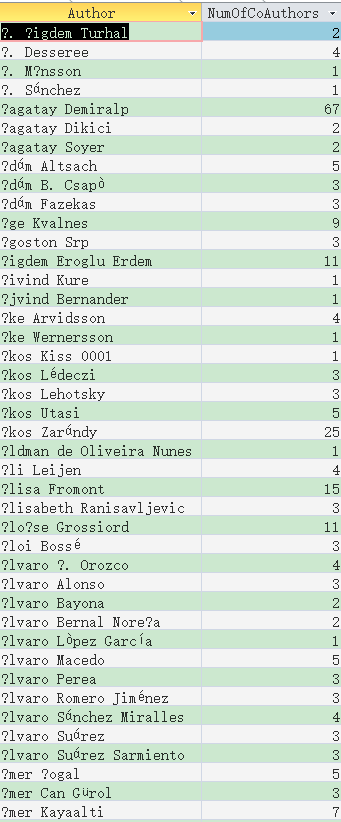
SELECT [查询NumOfCoAuthors].Author AS Author, COUNT(\*) AS NumOfCoAuthors

INTO NumOfCoAuthors

FROM [查询NumOfCoAuthors]

GROUP BY [查询NumOfCoAuthors].Author

result：total 127758 records



And then integrate all the information into a Complex table, including the Frequent\_Patterns table, the FirstPubYear table and the NumOfCoAuthors table

* Complex table：

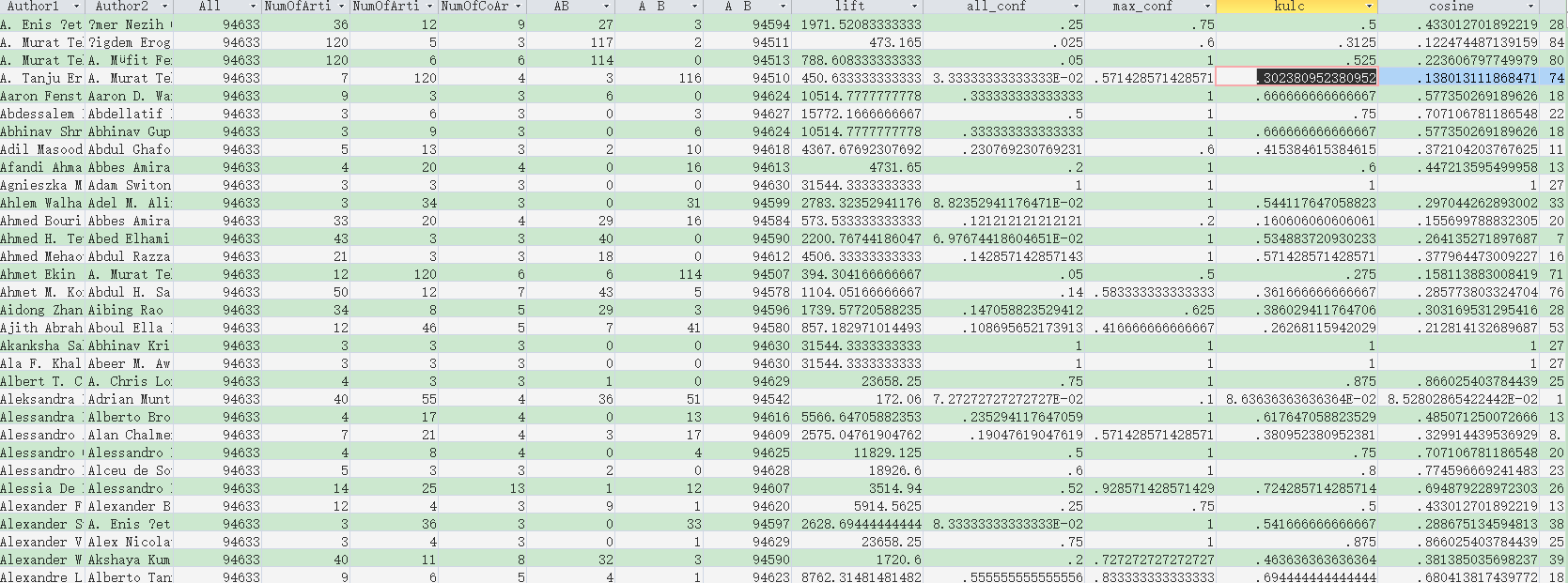
code：

SELECT Frequent\_Patterns.Author1 AS Author1, Frequent\_Patterns.Author2 AS Author2, Frequent\_Patterns.[All] AS [All], Frequent\_Patterns.[NumOfArticles1（A）] AS NumOfArticles1（A）, Frequent\_Patterns.[NumOfArticles2（B）] AS NumOfArticles2（B）, Frequent\_Patterns.[NumOfCoArticles（AB）] AS NumOfCoArticles（AB）, Frequent\_Patterns.[AB ̅] AS [AB ̅], Frequent\_Patterns.[A ̅ B] AS [A ̅ B], Frequent\_Patterns.[A ̅ B ̅] AS [A ̅ B ̅], Frequent\_Patterns.lift AS lift, Frequent\_Patterns.all\_conf AS all\_conf, Frequent\_Patterns.max\_conf AS max\_conf, Frequent\_Patterns.kulc AS kulc, Frequent\_Patterns.cosine AS cosine, Frequent\_Patterns.[χ^2] AS [χ^2], Frequent\_Patterns.IR AS IR, NumOfCoAuthors1.NumOfCoAuthors AS NumOfCoAuthors1, NumOfCoAuthors2.NumOfCoAuthors AS NumOfCoAuthors2, FirstPubYear1.FirstPubYear AS FirstPubYear1, FirstPubYear2.FirstPubYear AS FirstPubYear2 INTO Complex

FROM Frequent\_Patterns, NumOfCoAuthors AS NumOfCoAuthors1, NumOfCoAuthors AS NumOfCoAuthors2, FirstPubYear AS FirstPubYear1, FirstPubYear AS FirstPubYear2

WHERE Frequent\_Patterns.Author1=NumOfCoAuthors1.Author AND Frequent\_Patterns.Author2=NumOfCoAuthors2.Author AND Frequent\_Patterns.Author1=FirstPubYear1.Author AND Frequent\_Patterns.Author2=FirstPubYear2.Author;

result：total 24633 records



Lastly, get the more likely teacher-student relationship table by filtering the Complex table

* Complex\_Filtered table：

code：

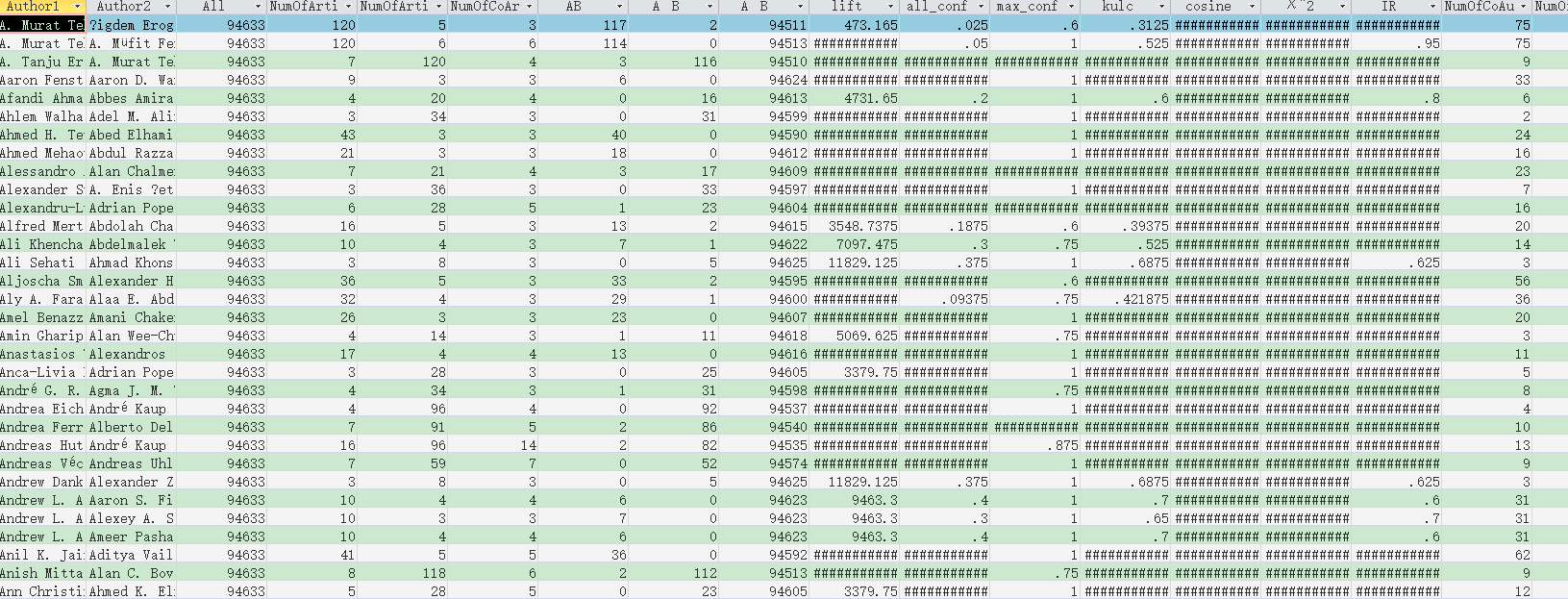
SELECT Frequent\_Patterns.Author1 AS Author1, Frequent\_Patterns.Author2 AS SELECT \*

INTO Complex\_Filtered

FROM Complex

WHERE max\_conf>0.5 AND (max\_conf-all\_conf)/all\_conf>0.5 AND IR>0.5 AND ABS(FirstPubYear1-FirstPubYear2)>5 AND ABS(NumOfCoAuthors1-NumOfCoAuthors2)>2

result：total 5951 records



Get the Advisor\_Advisee table by connecting Complex\_Filtered table 、Article\_Author table、Articles table

* Advisor\_Advisee table：

code：

SELECT IIf(Complex\_Filtered.FirstPubYear1<Complex\_Filtered.FirstPubYear2, Complex\_Filtered.Author1, Complex\_Filtered.Author2) AS Advisor, IIf(Complex\_Filtered.FirstPubYear1<Complex\_Filtered.FirstPubYear2, Complex\_Filtered.Author2, Complex\_Filtered.Author1) AS Advisee

FROM Complex\_Filtered

SELECT [查询Advisor\_Advisee].Advisor AS Advisor, [查询Advisor\_Advisee].Advisee AS Advisee,MIN(Articles.year\_) AS FirstCoPubYear, MAX(Articles.year\_) AS LastCoPubYear

INTO Advisor\_Advisee

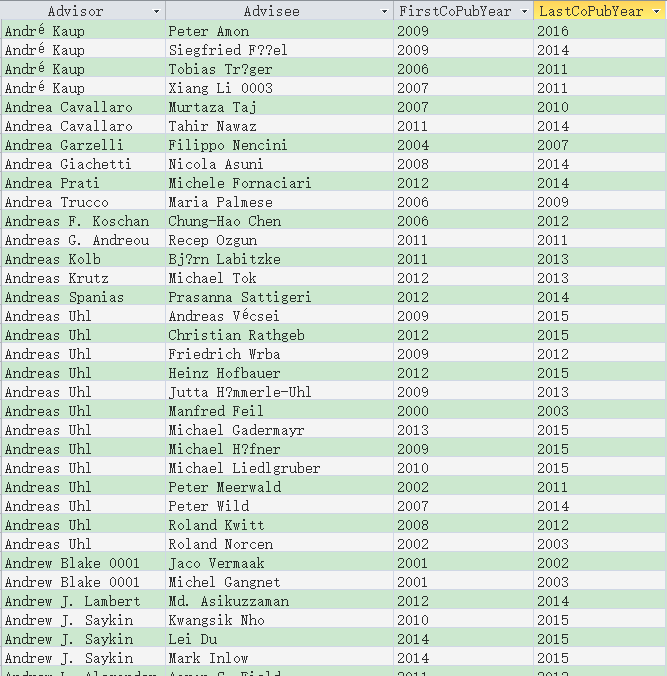
FROM [查询Advisor\_Advisee], Article\_Author AS Article\_Author1, Article\_Author AS Article\_Author2, Articles

WHERE [查询Advisor\_Advisee].Advisor=Article\_Author1.Author AND [查询Advisor\_Advisee].Advisee=Article\_Author2.Author AND Article\_Author1.title=Article\_Author2.title AND Articles.title=Article\_Author1.title

GROUP BY Advisor,Advisee

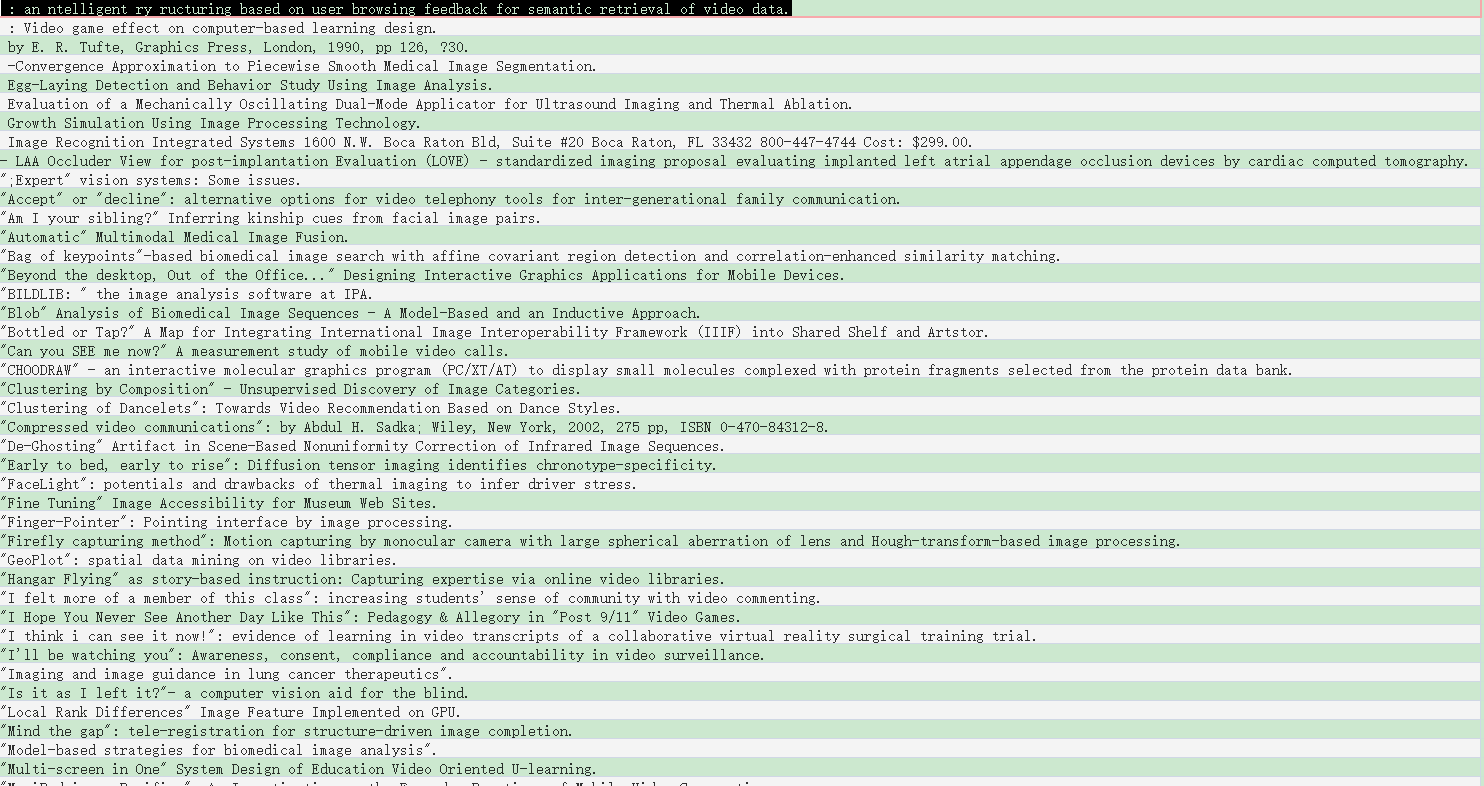
ORDER BY Advisor

result：total 5951 records



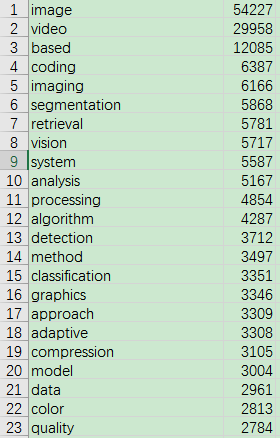
* Advanced Semantic Annotations

Firstil, extract the title of all the articles from the Article\_Author table



Then, get the frequent occurrence of meaningful academic keywords and their count by analysis the text information and sieve to "and", "on", "for" adverbs.

Result: total 5061 records.



The top 100 keywords are extracted as a vector base.

Calculate the semantic feature vector of each word and each author. And these 100 feature keywords appear simultaneously in an article, frequency and frequent patterns