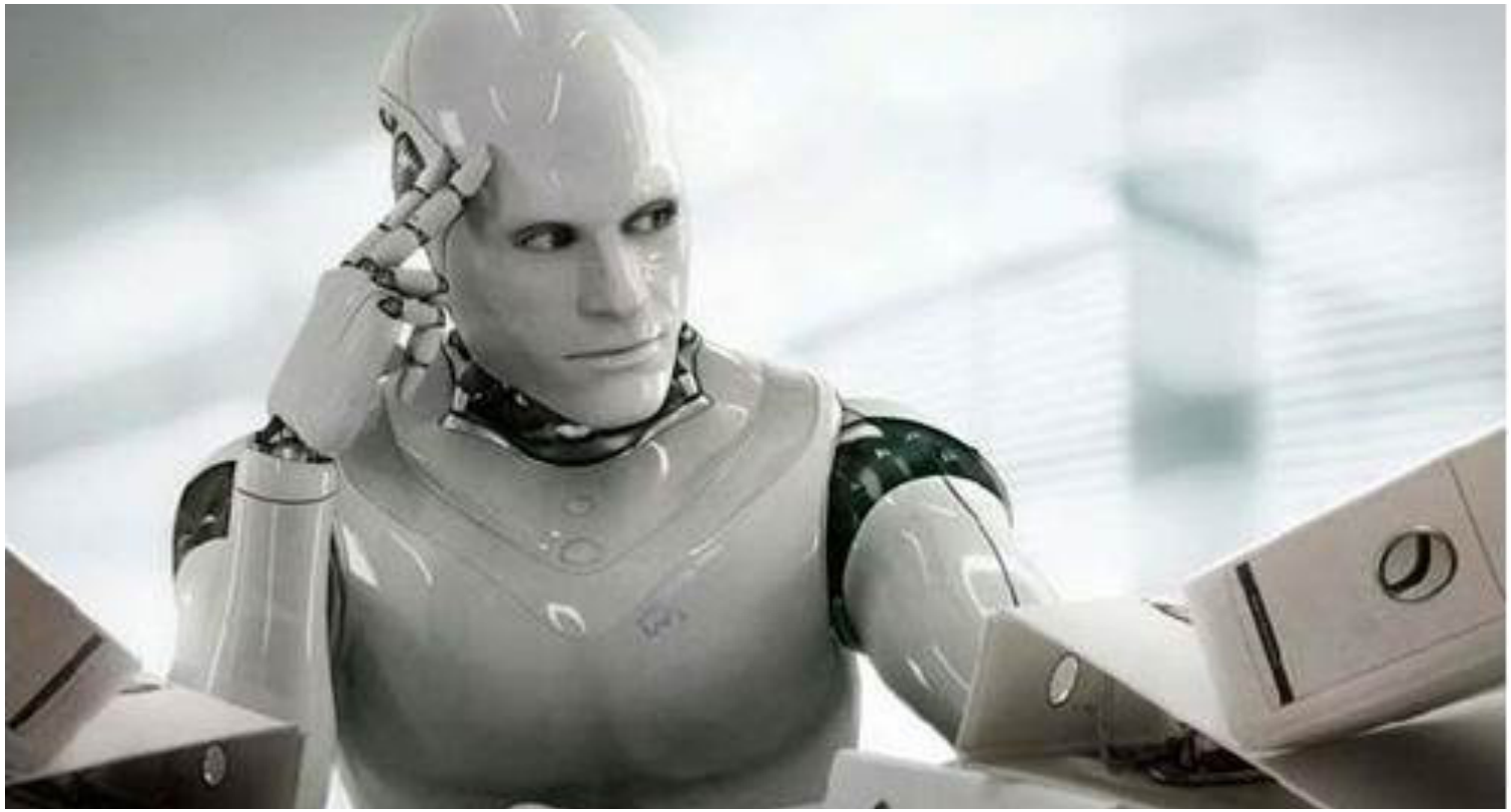
Relational Database Based on Natural Language

TEAM : CS Engineers

Members: H.ZHANG
YC.YANG
K.CHEN
K.SHI
X.LI

SQL is a good formal language for robots, but not for human beings!



Cited: http://blog.csdn.net/malefactor/article/details/50436735

Review of our past work

1. Propose NL to SQL preliminary algorithm.

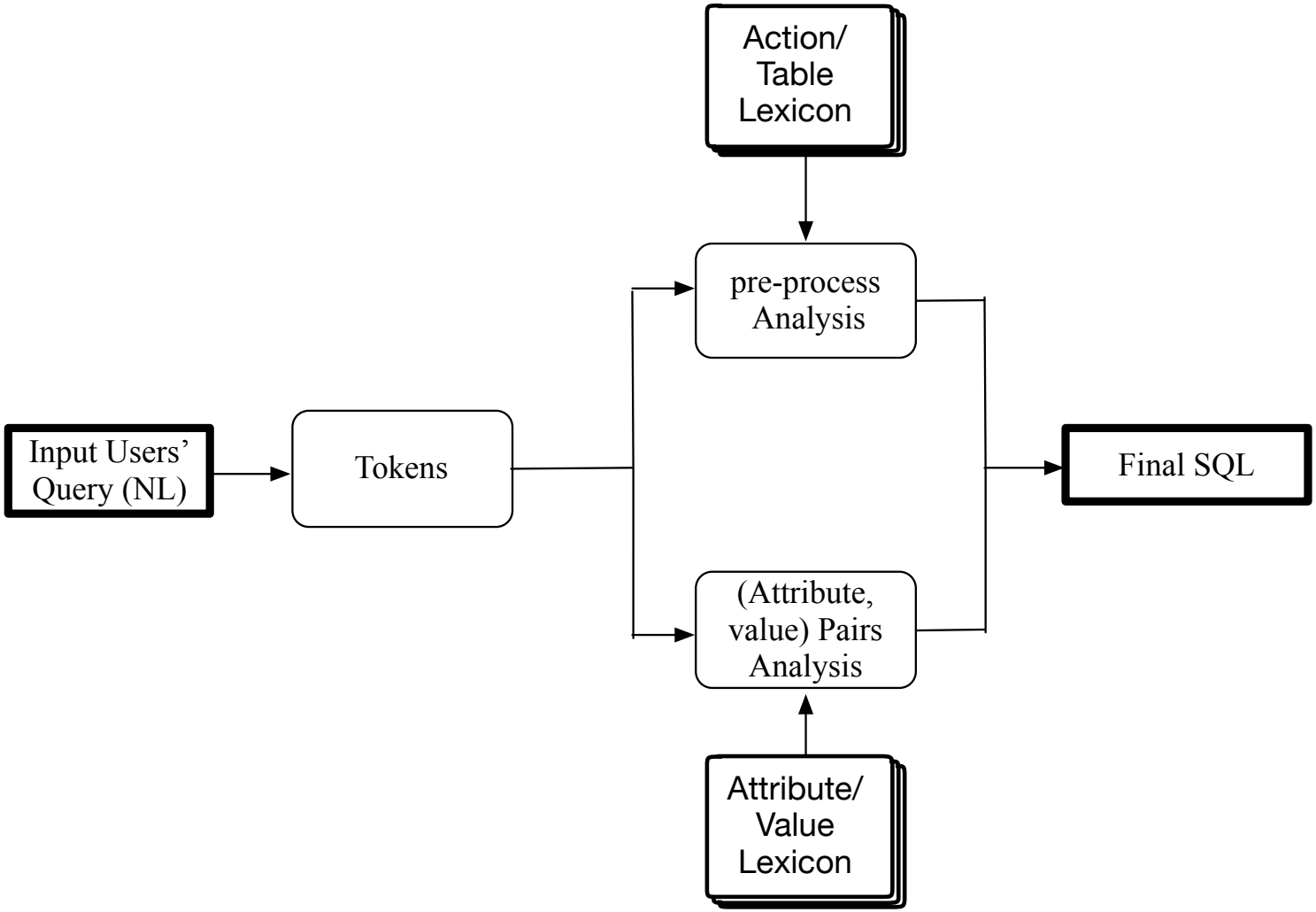
2. Build lexicons for all attributes and values, and do completely mapping

Unfinished work

- Decrease running time:
  - if databases have many attributes and values

- Function deficiency:
  - range searching
  - multiple conditions

Precise-Mapping Algorithm



Improved Algorithm

Attribute 1 (Primary key)	Attribute 2	Attribute 3	…
…	…	…	…

- Feature
  - Method of exclusion is used to find the values in primary key.
- Functions
  - Condition: e.g. where Name="…" and Gender ="…"
  - Range: between, larger than …
  - Not : e.g. not larger than…
  - User feedback

Challenges:

- How to improve Efficiency?
  - Token filter
  - Partition of String and Number
  - Do not mapping Primary Key

- How to ensure Accuracy?
  - Accurate search for Non-primary keys
  - Use user's feedback

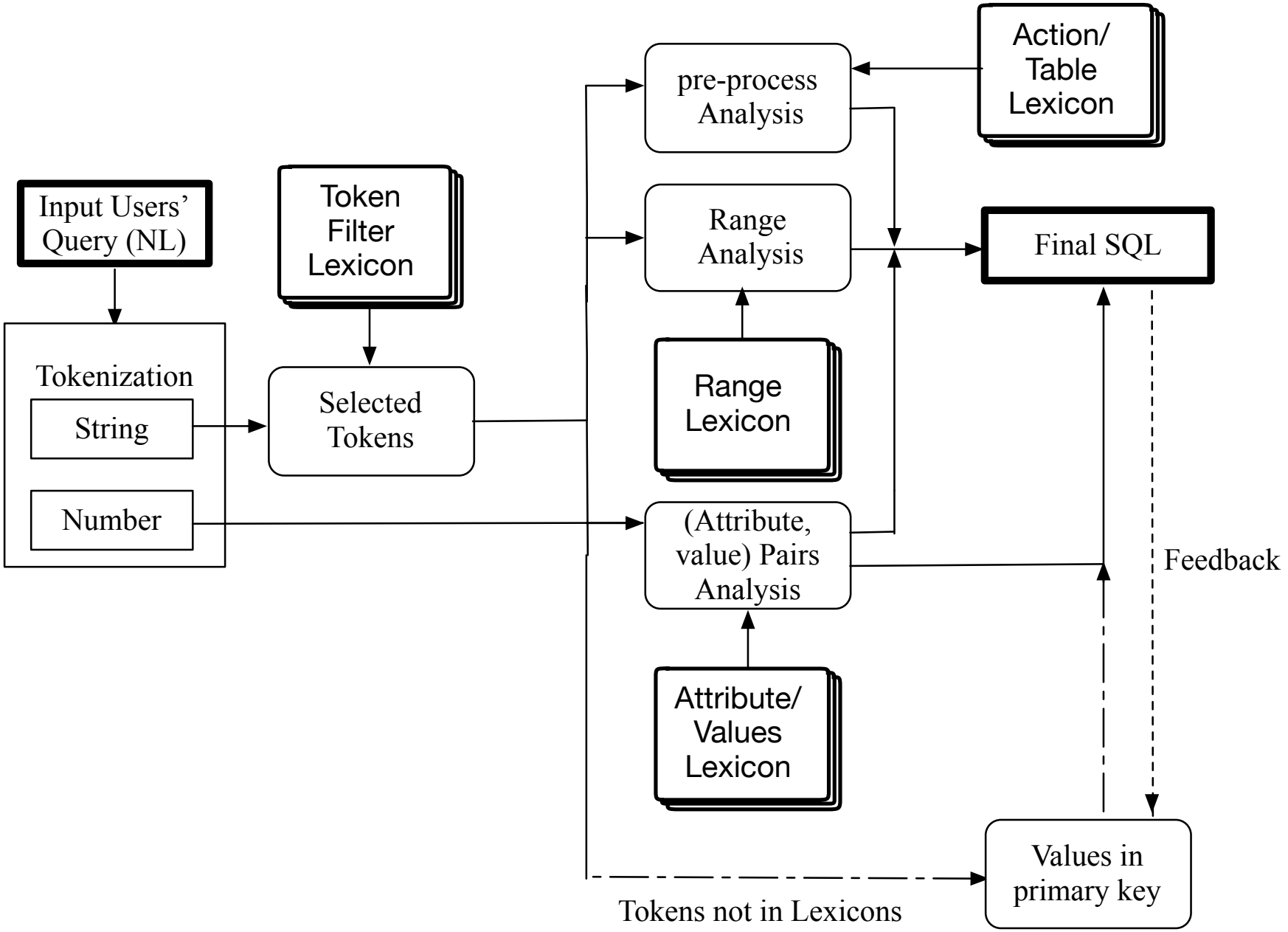
Lexicons:

Token Filter Lexicon : filter irrelevant words to get the **Selected Tokens.** E.g., is, are, find, do,…

Action/ Table Lexicon : mapping words related to actions/tables from **Selected Tokens.** E.g., select, delete…

Range Lexicon : mapping words related to Range from **Selected Tokens**. E.g., not, larger, small, between…

Attribute/ Value Lexicon : mapping words related to attributes (excluding primary key) and corresponding values from **Selected Tokens.** E.g., (Gender, male), (Dept., cs),…


Input Users' Query (NL)
Token Filter Lexicon
Tokenization
String
Number
Selected Tokens
pre-process Analysis
Action/ Table Lexicon
Range Analysis
Final SQL
Range Lexicon
(Attribute, value) Pairs Analysis
Feedback
Attribute/ Values Lexicon
Values in primary key
Tokens not in Lexicons

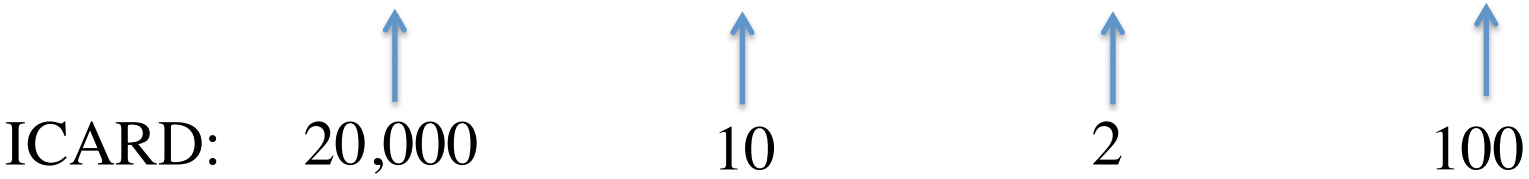
Tools:

➢Language: Java

➢IDE: Eclipse

Table

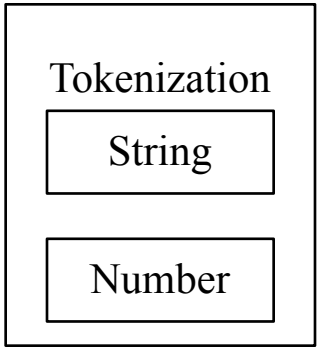
PRIM:Name	Department	Gender	Grade
Lily	CS	Female	80
Lucy	ECE	Female	70
Mike	BIO	Male	68
Jack	ECO	Male	90
…	…	…	…



Example 1(Range Select)

select student who is male and grade is not above 70

Tokenization



select student who is male and grade is not above 70

String:

Number:

String:
Selected Token

select student who is male and grade is not above

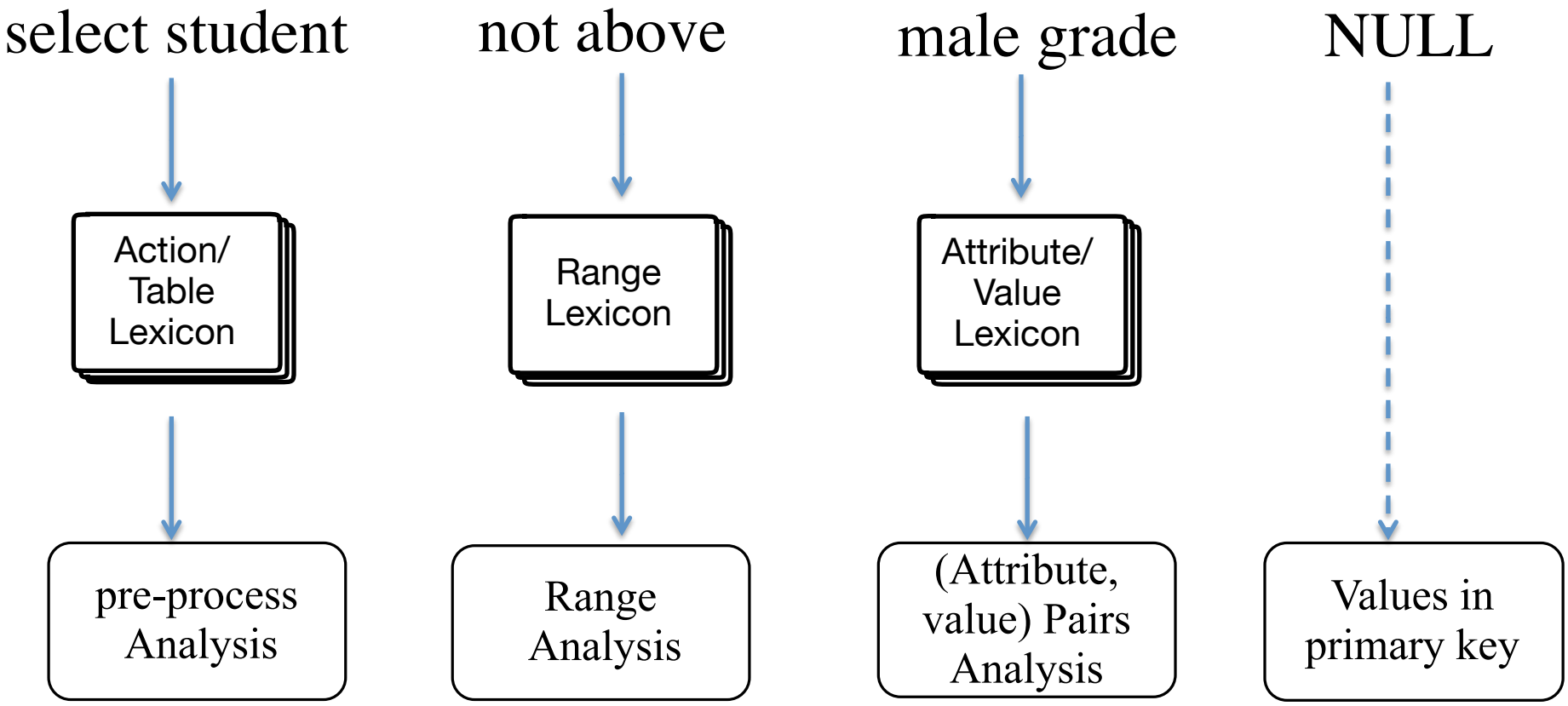
× × × ×

Selected Tokens

select student male grade not above

Selected Tokens:

select student name lucy grade not above



Form Query Syntax:

select student who is male and grade is not above 70

```
SELECT    X1, X2 …
FROM      TABLE
WHERE     Con1, Con2 …
```

Attribute	Value
PRIM: name	NULL
(gender)	male
grade	70

Range Analysis → Not larger : <=

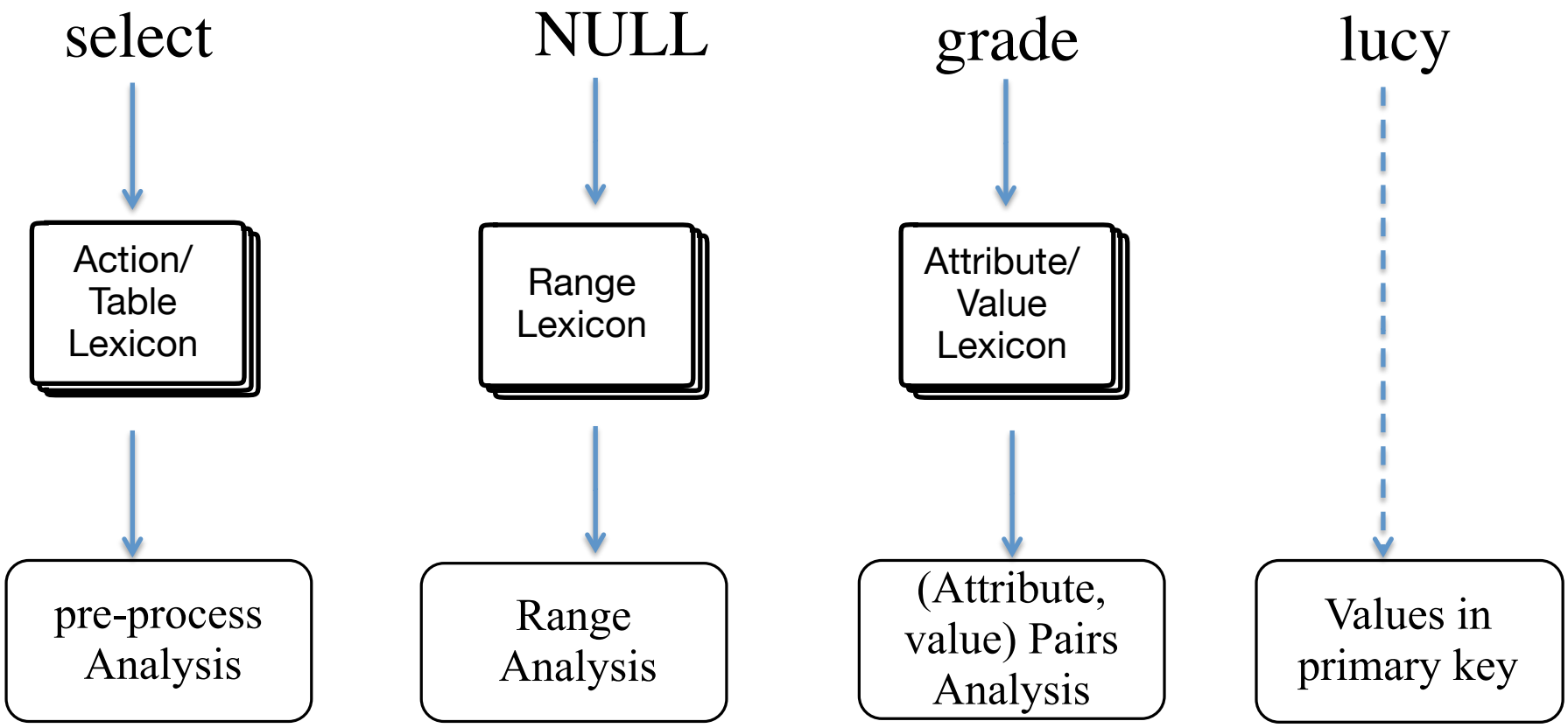
```
SELECT    name
FROM      student
WHERE     gender = male AND grade <= 70
```

Example 2(Value in Primary Key)

select	grade	whose	name	is	Lucy
		×		×	

Selected Tokens:

select student name lucy grade not above



Form Query Syntax: select grade whose name is Lucy

```
SELECT   X1, X2 …
FROM     TABLE
WHERE    Con1, Con2 …
```

```
SELECT    grade
FROM      student
WHERE     name = lucy
```

Attribute	Value
PRIM: name	lucy
grade	NULL

Advantage of Project

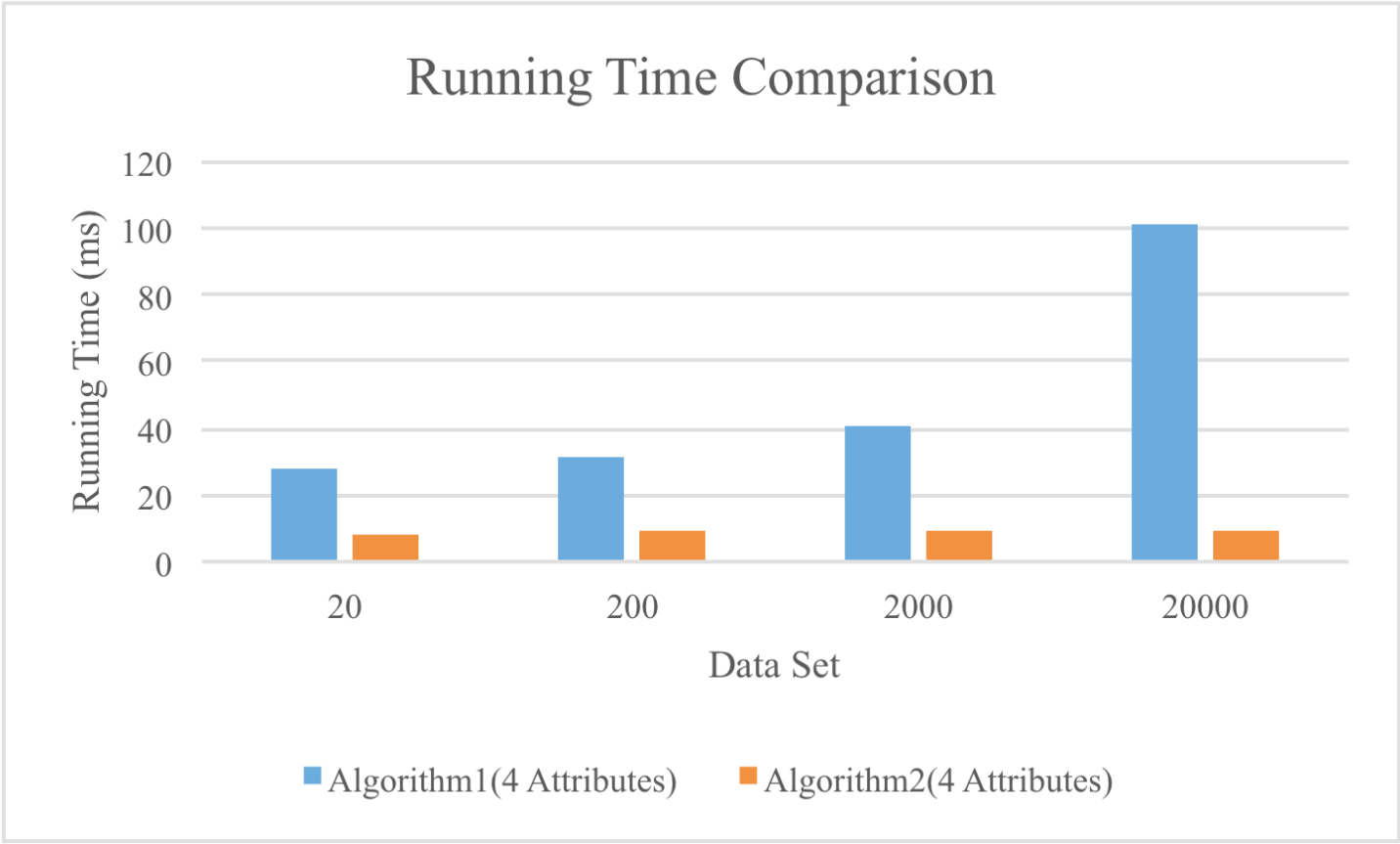
- Break the limitation:
  - Traditional data base cannot interact with NL
  - Our API makes it more intelligent

- Extendable and Portable:
  - Easy to support Chinese, Spanish,…
  - Easy to support different types of databases

- More efficient and more functional:
  - Do not need to match the primary key
  - Support range selection, Etc.

Running Time Comparison

Data	Algorithm1(4 Attributes)	Algorithm2(4 Attributes)
20	27.531414	8.594459
200	30.876006	8.906872
2000	40.438085	9.087095
20000	101.220836	9.310834



Future Work

- Fault Tolerance:
  - Can recognize spelling mistakes: “studen”
  - Can recognize grammar mistakes: “what the grade is Tom?”

- Token Filter Lexicon Improvement:
  - Word Parsing: Alien words
  - Semantic Analysis: More similar to SQL query (what is grade of Tom -> grade of student with name Tom)

Quick Demo

Link:

https://www.youtube.com/watch?v=gVOU4LSH41g&feature=youtu.be

Reference

- [1] W. Woods, R. Kaplan, and B. Webber, "The Lunar Sciences Natural Language Information System", Final Report, Technical Report 2378, Bolt Beranek and Newman Inc., 1972.
- [2] Gerard Salton, “Automatic Text Processing”, Addison-Wesley Publishing Company, 1989.
- [3] Tsukamoto Yoshitaka. Research on application system of natural language (in Japanese). Sijoh University, Thesis, 2007.
- [4]] A.-M. Popescu, O. Etzioni, and H. Kautz, “Towards a Theory of Natural Language. Interfaces to Databases”, Processdings of the 8th international conference on Intelligent user interfaces –IUI ’03, pp. 149-157, 2003.
- [5] Fei Li, “Constructing an Interactive Natural Language Interface for Relational Databases”, Univ. of Michigan, Ann Arbor, 2015.
- [7] E. A. and O. D.o, “An Algorithm Solving Natural Language Query Execution Problems on Relational Database. ”. International Journal of Advanced Computer Science and Application *IJACSA*, Vol. 3, No. 10, pp. 169-175, 2012.
- [6] QUEPY. http://quepy.machinalis.com/

Thanks for your attention!

3 Minutes Q&A