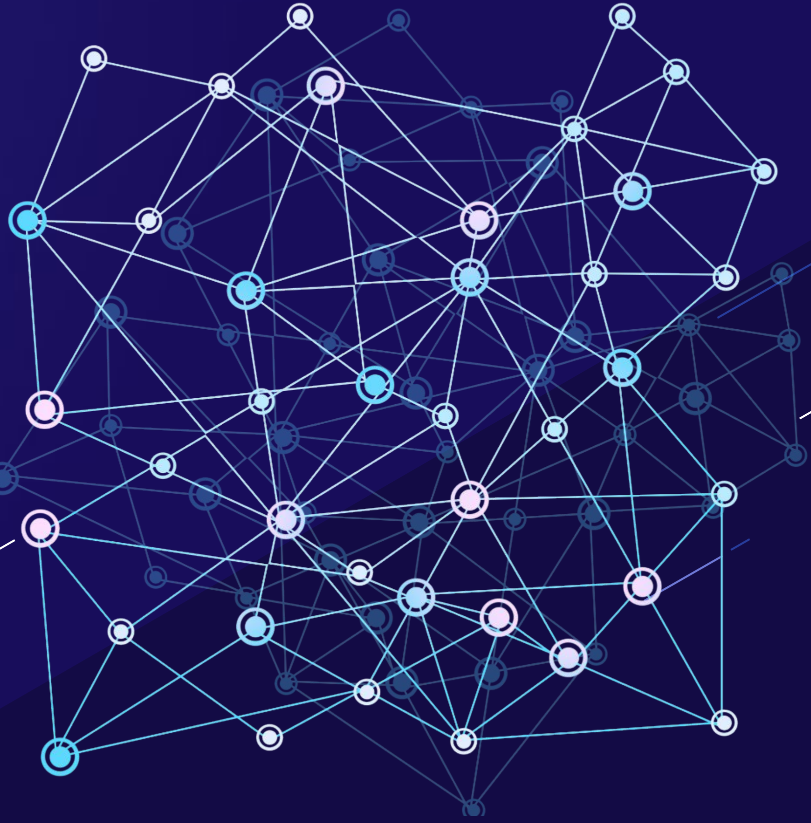
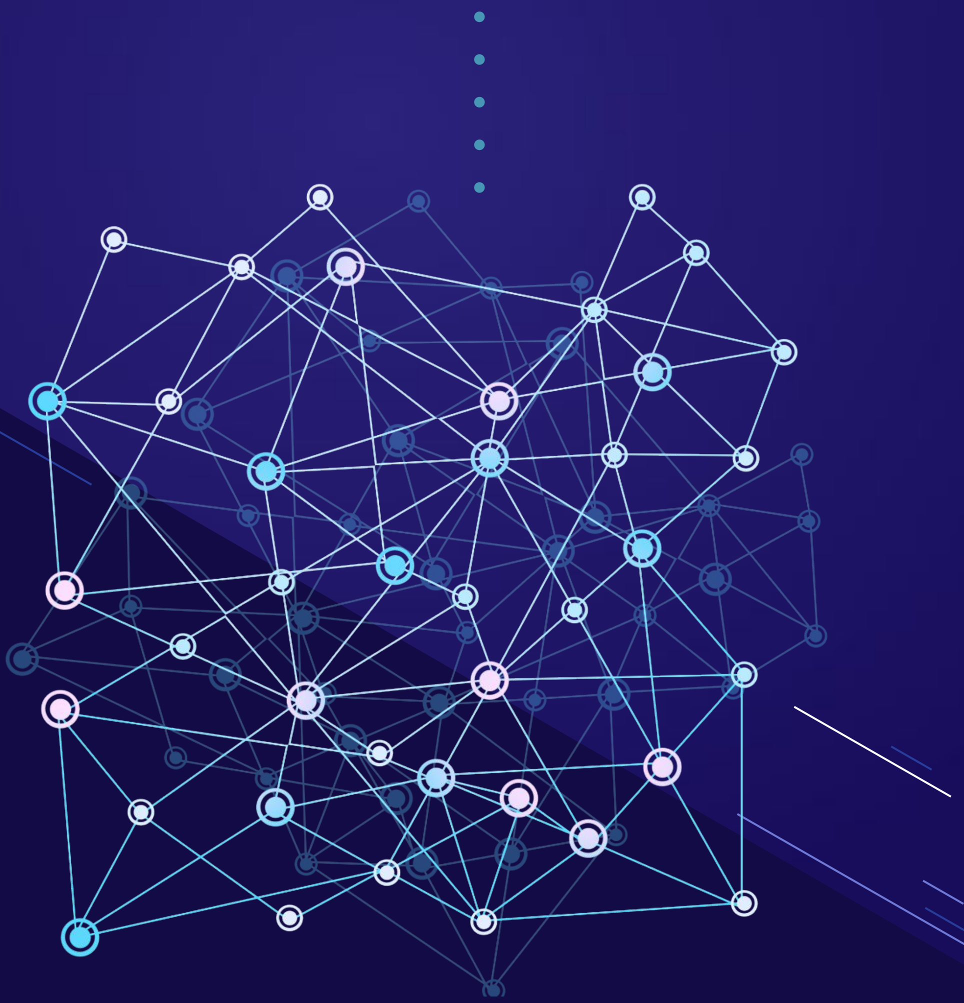
Effective Ways to select a Dataset from Large Corpus

여진영 교수님

김주찬, 김유진, Dobreva Iva



Contents



1 Subject

2 Graph Implementation

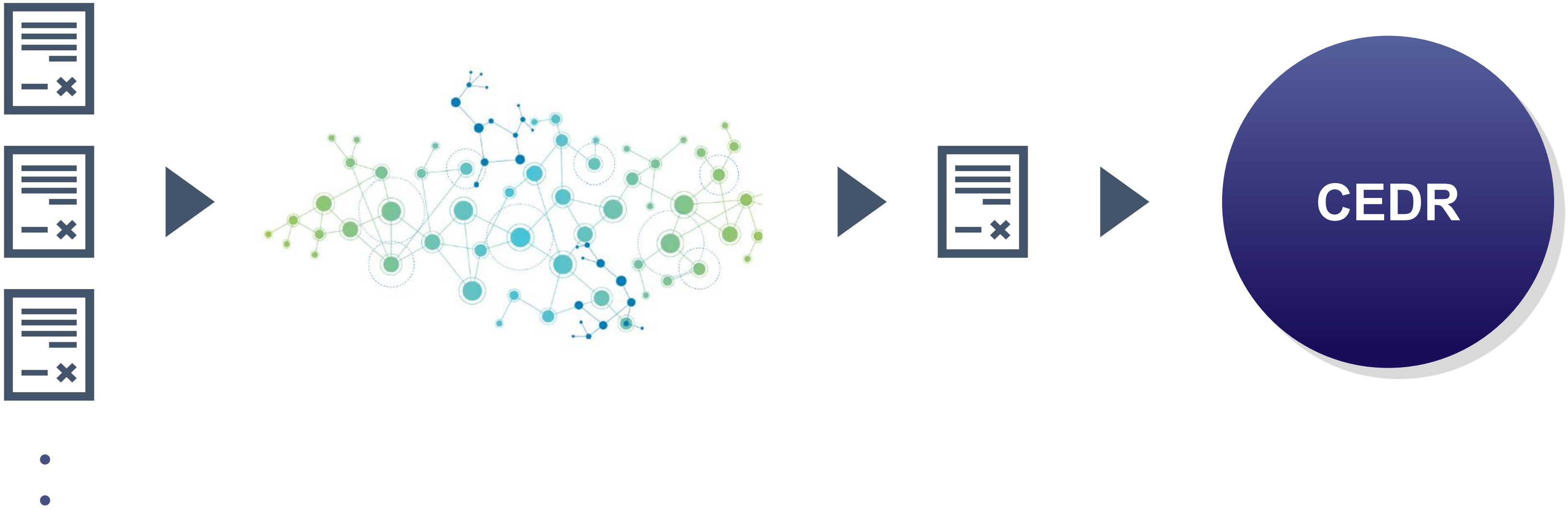
3 Influence Maximization

4 Training Result

Subject

Subje
ct

Data Selection with Influence Maximization



Subject

Process

Graph

Transform list data into graph

IMP

Influence Maximization with TIM

Training

Use seed dataset as a input of CEDR

Graph Implementation

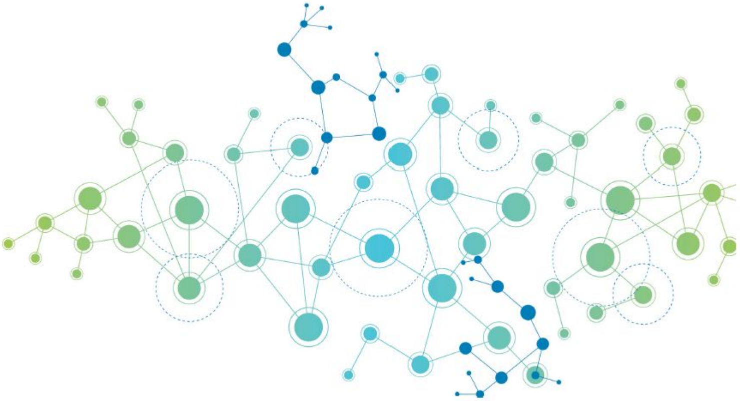
Structur
e

Node

- Document data (doc_id)
- Attributes: query_id, rel_id

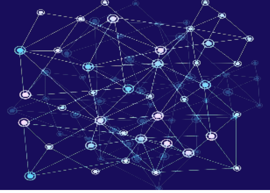
Edge

- Q(q);: all queries that both node participate in
- If n(Q) > 0 : connect the edge
- Weight: ∑ rel_id(q) / n(Q)



Rel_id:
1 if query and document is related

Graph Implementation



Implementation

Method 1

Step 1: create nodes
Step 2: calculate weight between
each nodes
Step 3: connect related nodes

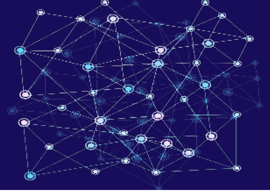
- Conclusion: takes several hours to be done (around 7 hours)

Method 2

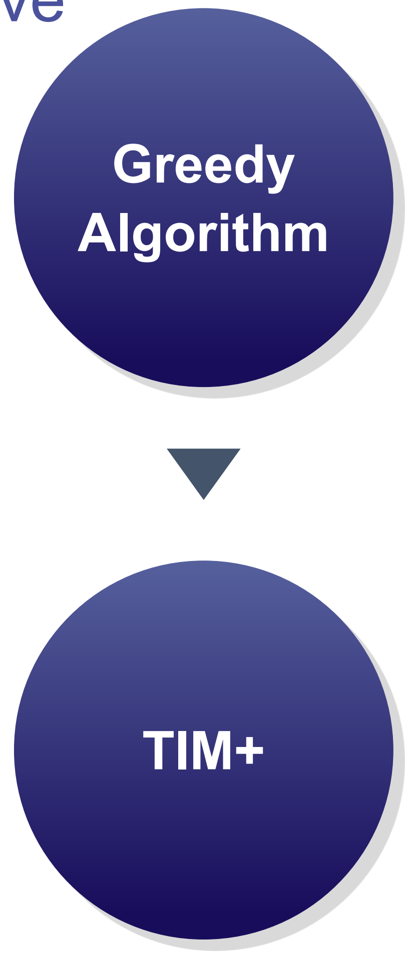
Step 1: create nodes
Step 2: make list of all edges
Step 3: connect related nodes at once

- Conclusion: failed in allocating memory for excessive amount of edges

Influence Maximization



How to solve



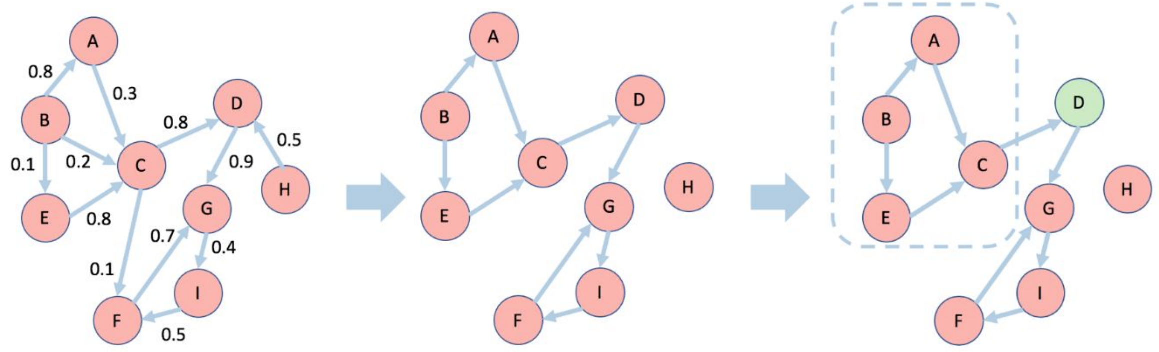
Kempe et al.

- the number of times influence function needs to be evaluated is quite huge
- selecting a seed set of size k with R number of MCS in a graph having n nodes and m edges
  -> $\boldsymbol{O(kmnR)}$ **evaluations**
- runs in days even when n and m are merely a few thousands

$O(kmn \cdot poly(\varepsilon^{-1})$ -> $O(k + l)(m + n)logn/\varepsilon^2$

Influence Maximization

Reverse Influence Sampling(RIS)



Reverse Reachable(RR) Set

- for a graph G
- generate graph g by removing each edge e according to its propagation probability $1 - p_e$
- for a node v take a set of nodes in g that can reach v

Generate a set R of many independent RR sets

Select k nodes to cover the maximum number of RR sets in R using the standard greedy algorithm

Influence Maximization

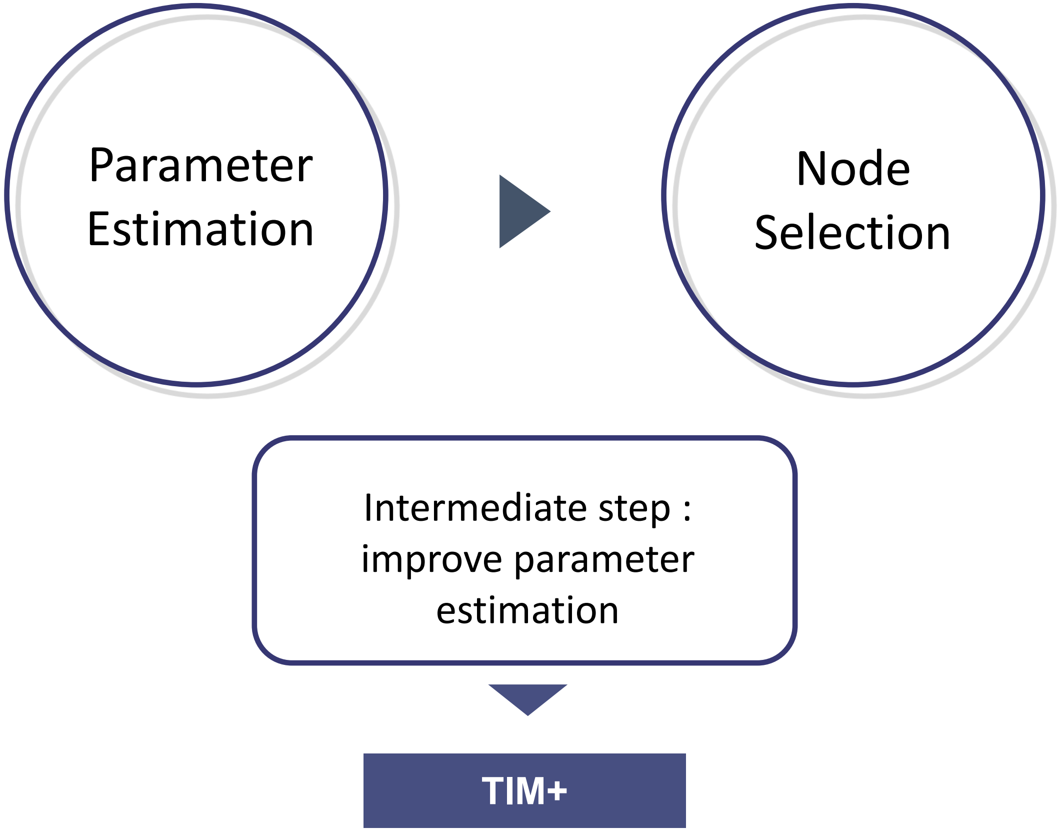
Two-phase Influence Maximization(TIM)

How many RR sets?

- **RIS**: count the total 'cost' of RR set construction and stop when total cost > a threshold

⇒ Significant computational overheads in practice

- **TIM**: bound the number of RR-set used



Parameter Estimation

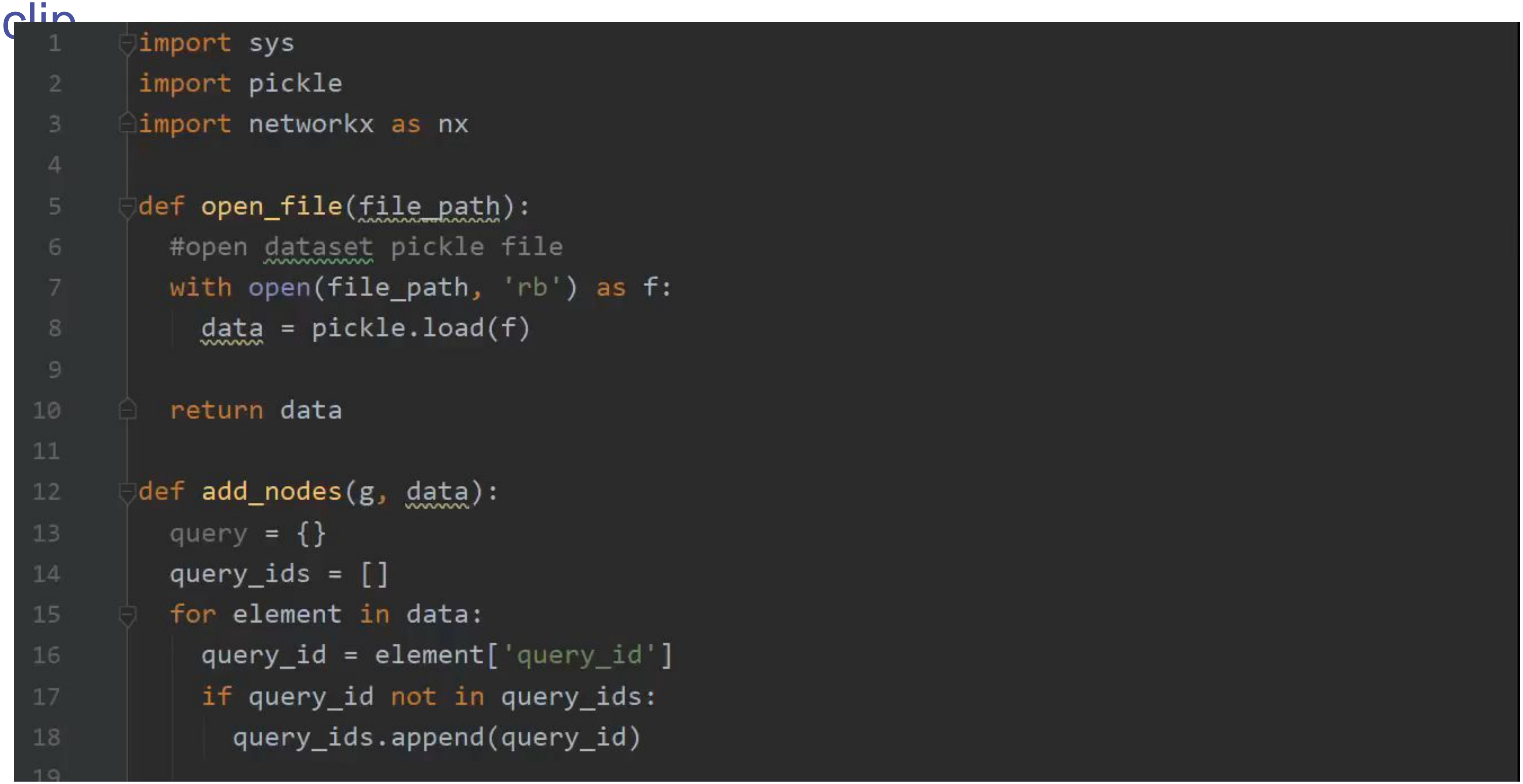
Node Selection

Intermediate step : improve parameter estimation

TIM+

Training Result

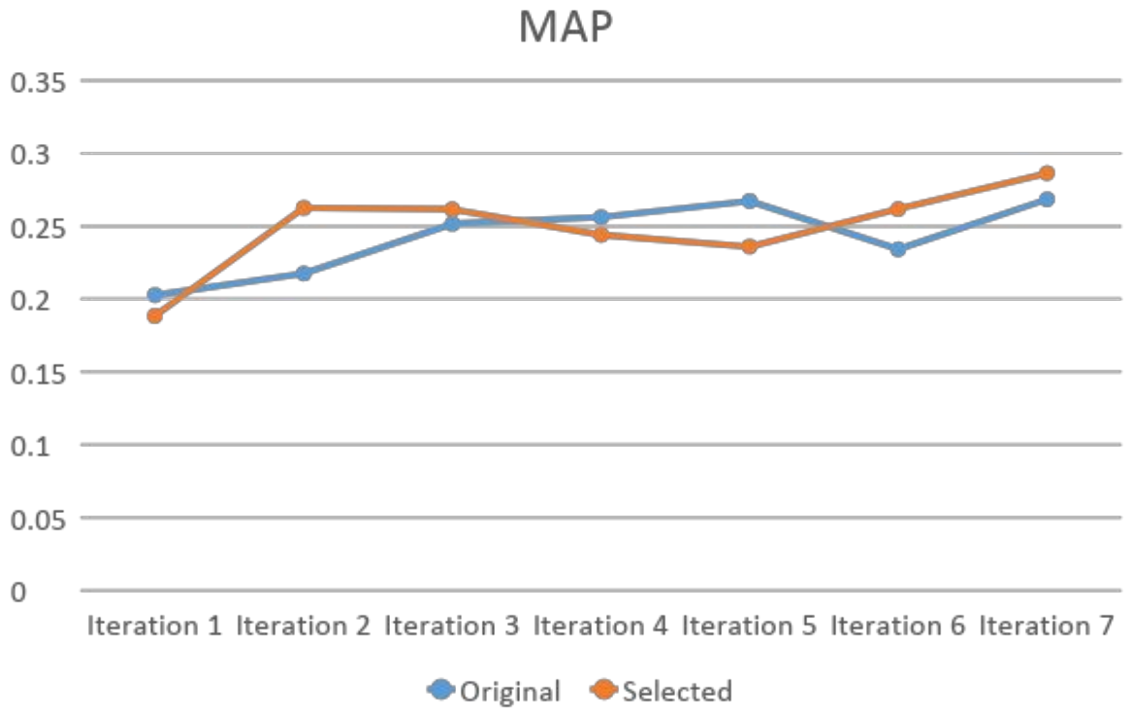
Video clip



```
import sys
import pickle
import networkx as nx

def open_file(file_path):
  #open dataset pickle file
  with open(file_path, 'rb') as f:
    data = pickle.load(f)

  return data

def add_nodes(g, data):
  query = {}
  query_ids = []
  for element in data:
    query_id = element['query_id']
    if query_id not in query_ids:
      query_ids.append(query_id)
```

Training Result

MAP comparison



Result

- dataset reduction:
  110,000 -> 50,000
- training time decreased:
  11 hours -> 6 hours
- accuracy improved

THANK YOU

감사합니다