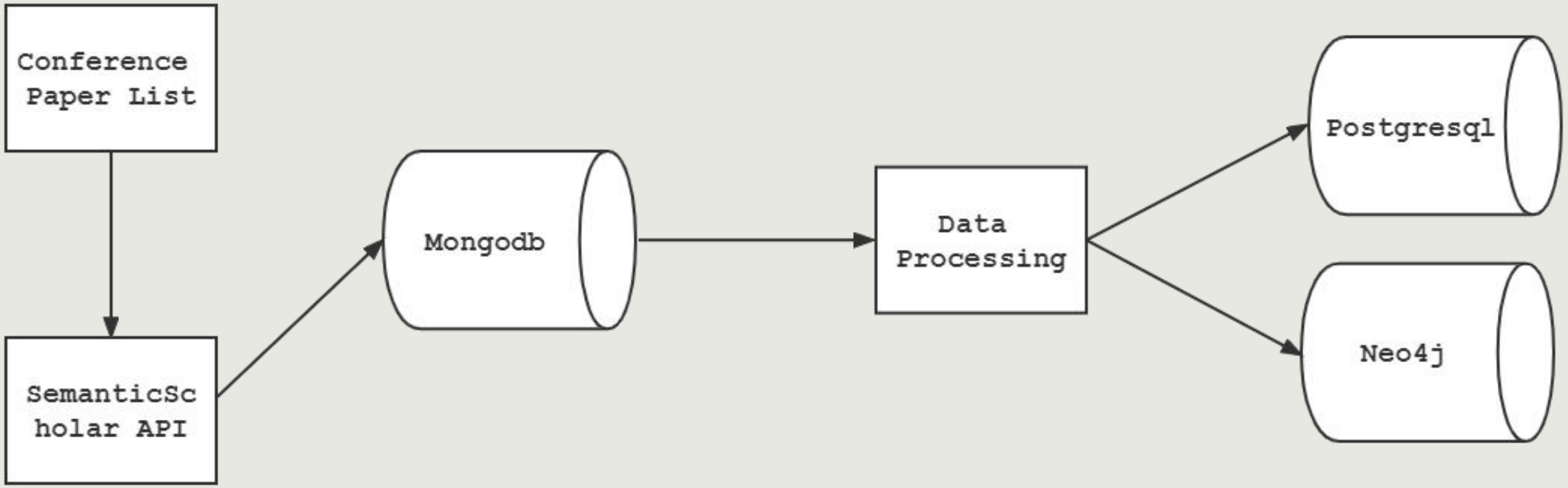
Recent NLP Area Analysis with Postgresql and Neo4j

Dawei Li, Ruihan Wang, Jinhe Wen

2022.12.8

Overview


Conference Paper List
SemanticScholar API
Mongodb
Data Processing
Postgresql
Neo4j

Data Crawling

- Dataset: NLP Paper Dataset
  - Papers in the three top conference(ACL, EMNLP, NAACL) of NLP area
  - ACL Anthology – ACL Anthology
- SemanticScholar API
  - Semantic Scholar Academic Graph API | Semantic Scholar
  - paper search, paper lookup, author lookup …
- Crawling
  - Scrapy
  - Mongodb


```
_id: ObjectId("636b2be85a3c094b864d5943"),
paperId: '6ebfbc954b9975d2f2651f380b9bdf46ae963178',
title: 'PLATO: Pre-trained Dialogue Generation Model with Discrete La
abstract: 'Pre-training models have been proved effective for a wide
us kinds of conversations, including chit-chat, knowledge grounded dial
context and the uni-directional characteristic of language generation.
of response generation and latent act recognition are designed and car
superiority of the proposed framework.',
venue: 'ACL',
year: 2019,
referenceCount: 37,
citationCount: 139,
influentialCitationCount: 21,
publicationDate: '2019-10-17',
authors: [
  {
    authorId: '2026806395',
    name: 'Siqi Bao',
    aliases: null,
    affiliations: [],
    homepage: null,
    paperCount: 18,
    citationCount: 276,
    hIndex: 7
  },
  {
    authorId: '46350360',
    name: 'H. He',
    aliases: [ 'Huan He', 'Huang He' ],
    affiliations: [],
    homepage: null,
    paperCount: 17,
    citationCount: 283,
    hIndex: 8
  },
  {
    authorId: '2145903238',
    name: 'Fan Wang',
    aliases: null,
    affiliations: [],
    homepage: null,
    paperCount: 35,
    citationCount: 506,
    hIndex: 10
  },
  {
    authorId: '40354707',
    name: 'Hua Wu',
    aliases: null,
    affiliations: [],
    homepage: null,
    paperCount: 169,
    citationCount: 7243,
    hIndex: 40
  }
],
citations: [
  {
    paperId: '5b8f1450584332cb81638b5823d48f3b632af511',
    citationCount: 0,
    influentialCitationCount: 0
  },
```

Keyword Extraction

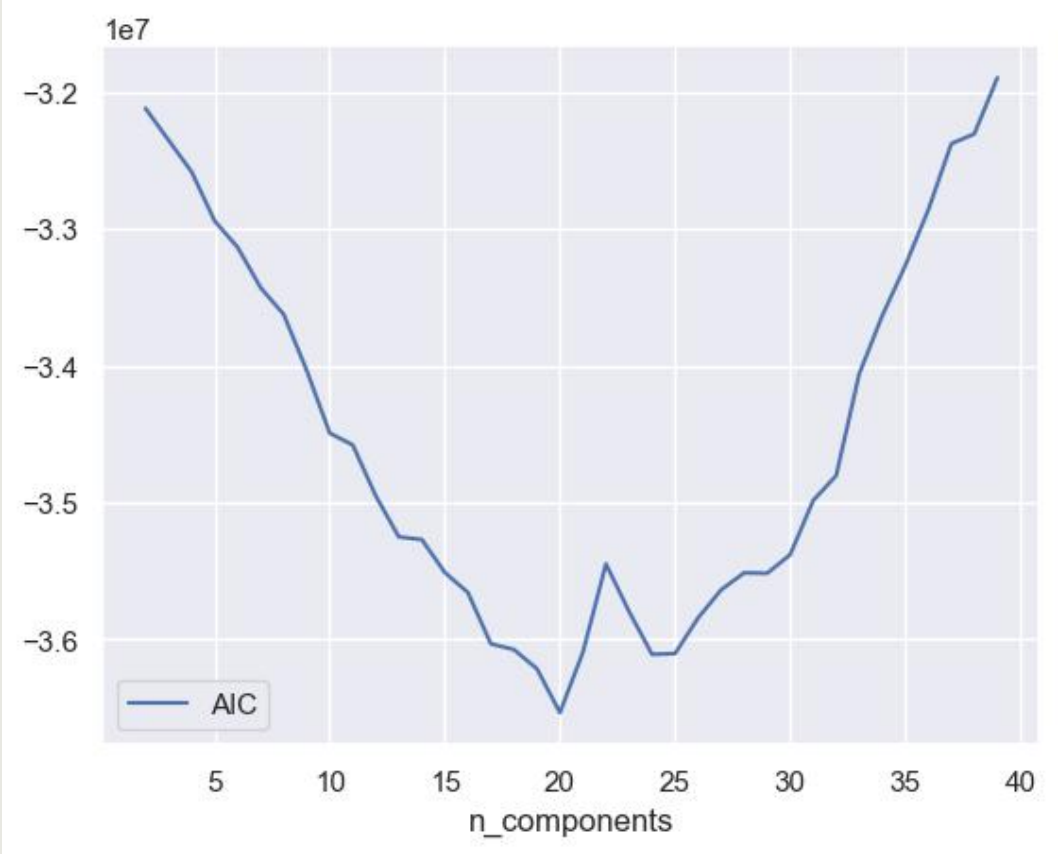
- Extract keywords from abstracts
  - KeyphraseVectorizers package in Python
  - Filter the extracted keywords

```
Abstract:
Existing commonsense knowledge bases often organize tuples in an isolated manner, which is deficient for commonsense conversational models t
o plan the next steps. To fill the gap, we curate a large-scale multi-turn human-written conversation corpus, and create the first Chinese c
ommonsense conversation knowledge graph which incorporates both social commonsense knowledge and dialog flow information. To show the potent
ial of our graph, we develop a graph-conversation matching approach, and benchmark two graph-grounded conversational tasks.

Keywords:
[('first chinese commonsense conversation knowledge graph', 0.8351), ('conversation corpus', 0.6502), ('conversation matching approach', 0.5
647), ('conversational models', 0.5639), ('commonsense knowledge bases', 0.5452)]
```

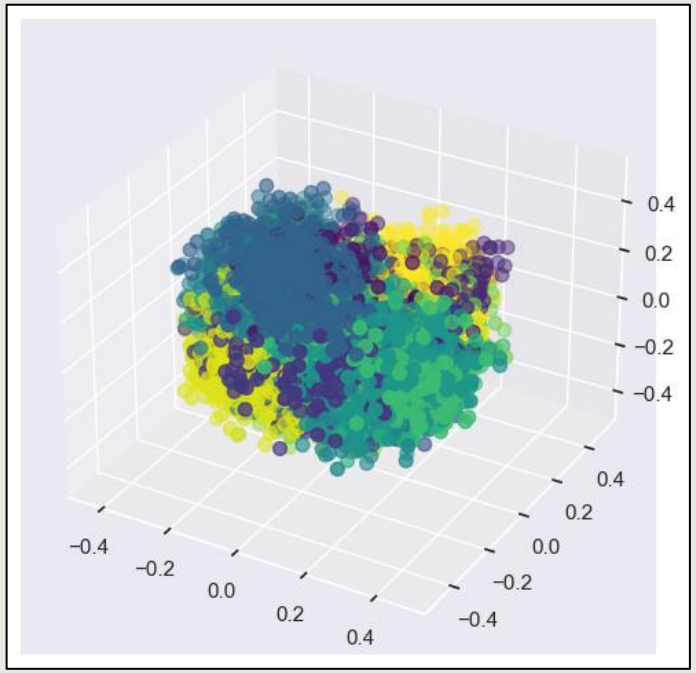
Area Extraction

- Clustering keywords to get research areas
  - Sentence–transformer to encode each keyword
  - Gaussian Mixture Model (GMM)
  - AIC matrix to decide the best clustering number



Area Extraction

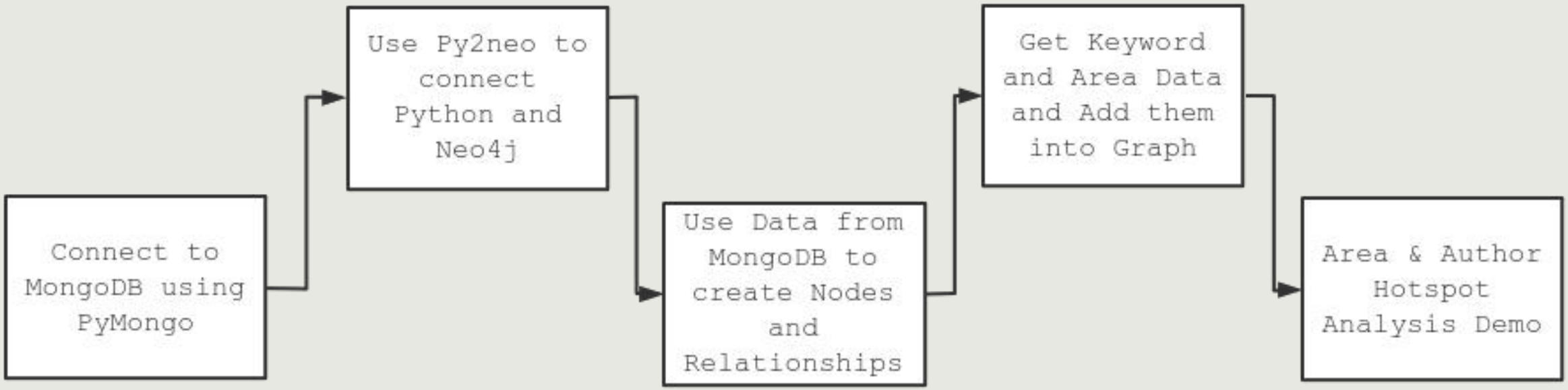
- Clustering result visualization




```
['fluent response generation',
 'neural dialogue systems',
 'chat style dialogue',
 'persuasion conversations',
 'conversation goals',
 'effective dialogue representations',
 'bot conversations',
 'dialog responses',
 'training dialogue policies',
 'augmented dialogues',
 'shot dialogue state tracking',
 'neural dialog models',
 'spoken dialog representations',
 'psychotherapy conversations',
 'utterance prediction',
 'speaker sensitive response evaluation model',
 'dialog dataset',
 'domain chatbots',
 'dialogue response ranking training',
```


Neo4j:
Py2neo
Focus on
Area Hotspot

Overview


Connect to MongoDB using PyMongo
Use Py2neo to connect Python and Neo4j
Use Data from MongoDB to create Nodes and Relationships
Get Keyword and Area Data and Add them into Graph
Area & Author Hotspot Analysis Demo

Nodes and Relationships

```
paper_node = Node('Paper',
                  # dbId=paper_data['_id'],
                  paperId=paper_data['paperId'],
                  title=paper_data['title'],
                  abstract=paper_data['abstract'],
                  venue=paper_data['venue'],
                  year=paper_data['year'],
                  referenceCount=paper_data['referenceCount'],
                  citationCount=paper_data['citationCount'],
                  influentialCitationCount=paper_data['influentialCitationCount'],
                  publicationDate=paper_data['publicationDate']
                  )
self.g.create(paper_node)
```

```
rel = "wrote"
paper_author_rel = Relationship(author_node, rel, paper_node, rank=i+1)
self.g.create(paper_author_rel)
```

Plugin: apoc

nodeLabel	nodeCount
Paper	33441
Author	7657
Area	17
Keyword	7433
Affiliation	619

labelCount	5
relTypeCount	6
propertyKeyCount	29
nodeCount	49167
relCount	128018

nodeRel	relCount
(:Author)-[:works in]->()	1327
()-[:belongs to]->(:Area)	7433
(:Author)-[:cooperate]->()	26071
(:Keyword)-[:belongs to]->()	7433
()-[:belongs to]->()	7433
()-[:is about]->()	9819
()-[:cited by]->(:Paper)	70865
(:Paper)-[:cited by]->()	70865
(:Paper)-[:is about]->()	9819
()-[:cited by]->()	70865
()-[:cooperate]->(:Author)	26071
()-[:works in]->(:Affiliation)	1327
()-[:is about]->(:Keyword)	9819
()-[:wrote]->(:Paper)	12503
()-[:works in]->()	1327
()-[:cooperate]->()	26071
()-[:wrote]->()	12503
(:Author)-[:wrote]->()	12503

Area Hotspot: Count

```
def area_count(self, area):
    # for this keyword, how many papers are there
    cql = "MATCH (aff)<-[r2:`works in`]-(aut)-[r1:`wrote`]->(p)-->()-->(Area {area:'" + area + "'}) RETURN " \
                                        "count(DISTINCT p) AS cnt_p, " \
                                        "count(DISTINCT aut) AS cnt_aut, " \
                                        "count(DISTINCT aff) AS cnt_aff"
    affiliation_count = self.g.run(cql).to_data_frame().values
    # print(affiliation_count)
    return affiliation_count
```

Area Hotspot: from Authors and Citations

- Author Influence (paperCount, citation, hindex)
  - Rank: First author, second author,…, giving a weight
  - Sum & Mean
- Citation Influence (citation, influentialCiatation)

Area	PaperCount	Author_pap	Author_cit	Author_hIn	Cit_cit	Cit_inf
Society and Application	1269	44487.7	1832110.4	12876.4	299077	41518
Multilingual and Translation	952	41286.2	1887046.8	10763.1	297052	40843
Model Architecture	1043	38531.7	1653270.7	10717	451452	59929
Natural Language Understanding	710	29937.1	1087900.9	7709.4	248842	38991
Dialogue	729	28350.1	1104618.4	7269.5	104969	14930
Learning Paradigma	660	25733.5	976093	6794	122727	17700
Text Mining and Retrival	575	24039.1	788652	6009.1	123804	18050
Language Model	488	21576.3	875113.7	5360.2	203235	29959
Linguistics	534	20778.2	809326.2	5596.7	71087	9517
Relation Extraction	520	18215.6	578695.2	5031.9	86084	13132
Knowledge Graph	516	17492.2	540012.6	5000.6	88660	12034
Grammar and Syntax	324	14194.1	514027.8	3507.6	47074	7250
Multi-modal	320	13677.9	461104.7	3150.1	42579	5702
Text Generation	327	12792	468861.9	3592.4	53035	7455
Summarirzation	289	10505.8	539588.9	2979.1	166181	22417
Representation Learning	265	9411.2	322466.8	2491.9	49281	6720
Sentiment Analysis	298	9370.7	238960.9	2489.8	31401	3782

Who is the NLP Star ! ——Plugin: GDS ——Centrality

- **Project a graph**
- **Memory Estimation**
- **Stream**

Node: Author

Relationships: cooperate

Order by Score DESC

	nodeCount	relationshipCount	bytesMin	bytesMax	requiredMemory
0	7657	26071	185520	185520	181 KiB

	author	score
0	Ming Zhou	13.175026
1	Yejin Choi	12.532505
2	Luke Zettlemoyer	11.930771
3	Nanyun Peng	11.511896
4	D. Roth	9.931232
..	...	...
95	Clement Chung	3.557559
96	Shrimai Prabhumoye	3.554349
97	Veselin Stoyanov	3.533982
98	Paolo Papotti	3.532289
99	Percy Liang	3.526467

Postgresql:

Psycopg2
Focus on
Time Aggregation Analysis &
Potential Analysis

Postgresql Schema

- Design principle
  - Reduce redundant information
  - Highly coupled inside tables
  - Minimize join operations
- Tables
  - paperInfo
  - AuthorInfo

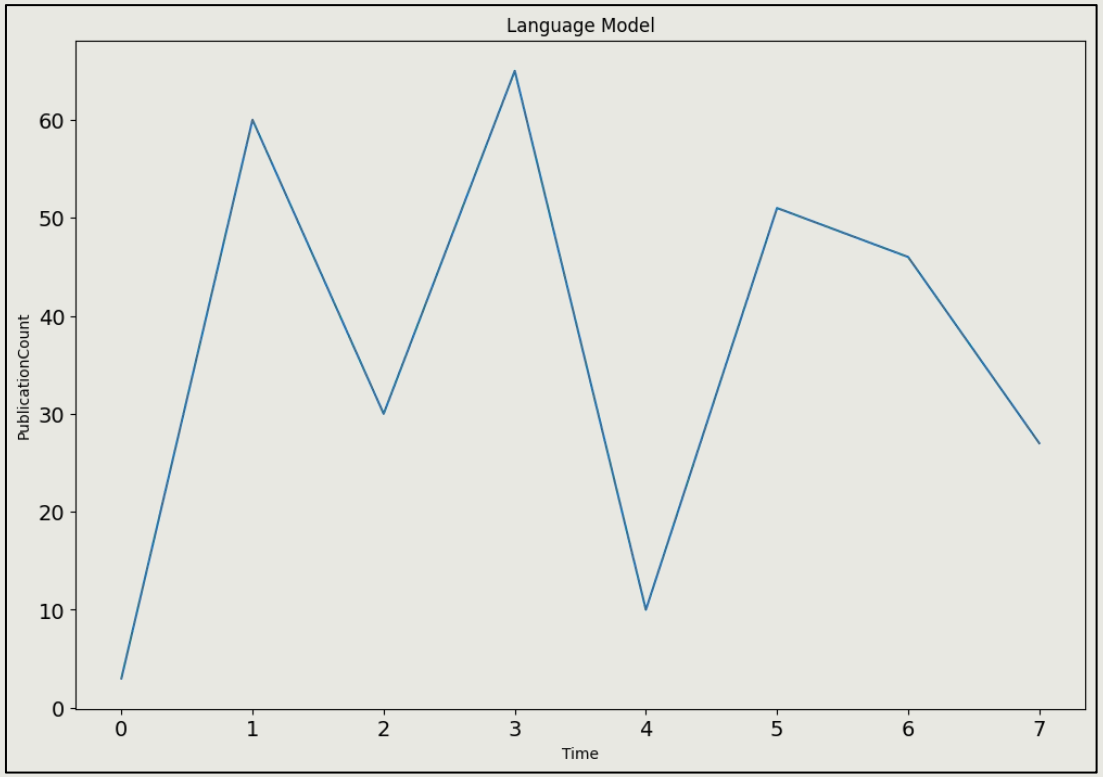
```
create table paperInfo(
    paperId    text    PRIMARY KEY,
    title    text,
    abstract    text,
    venue    text,
    year    integer,
    referenceCount    integer,
    citationCount    integer,
    influentialCitationCount    integer,
    publicationDate    date,
    authorsId    text [],
    keywords    text [],
    area    text []
);

create table authorInfo(
    authorId    text    PRIMARY KEY,
    name    text,
    aliases    text [],
    affiliations    text [],
    paperCount    integer,
    citationCount    integer,
    hIndex    integer
);
```

Time-related Area Analysis

- Analyze the change in the popularity of different areas over time
  - Time interval: quarterly
  - Metirc: paperCount

```
select year, q, count(*) from
    (select *,
    ceil((split_part(text(publicationDate),'-',2))::numeric/3)as q
    from paperInfo) as p
where 'Language Model' = any(area)
group by year, q
order by (year, q);
```



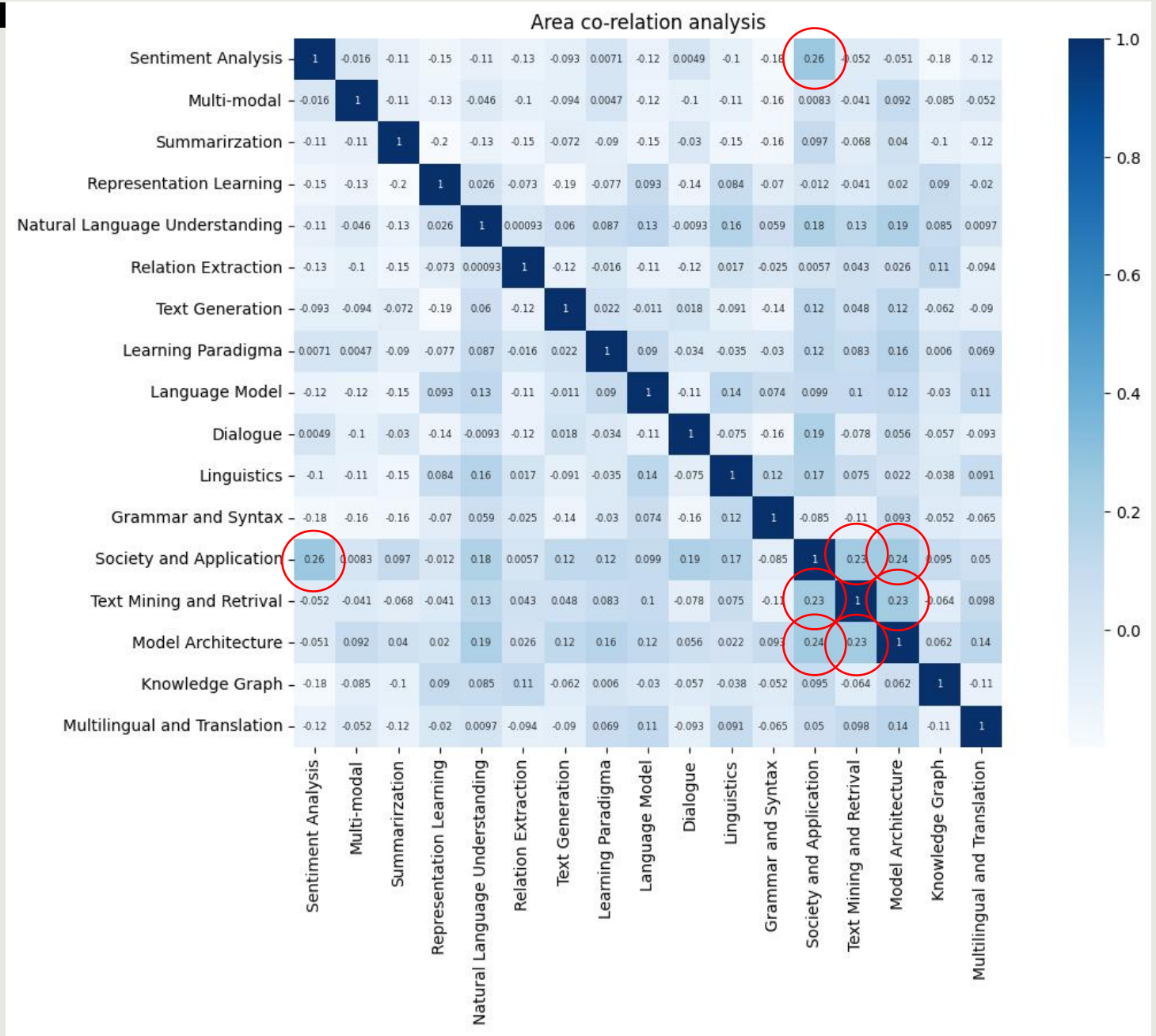
Area Correlation Analysis

- Analyze correlations between different areas
  - Paper: areas of the same paper have high correlation
  - Author: areas of the same author have high correlation

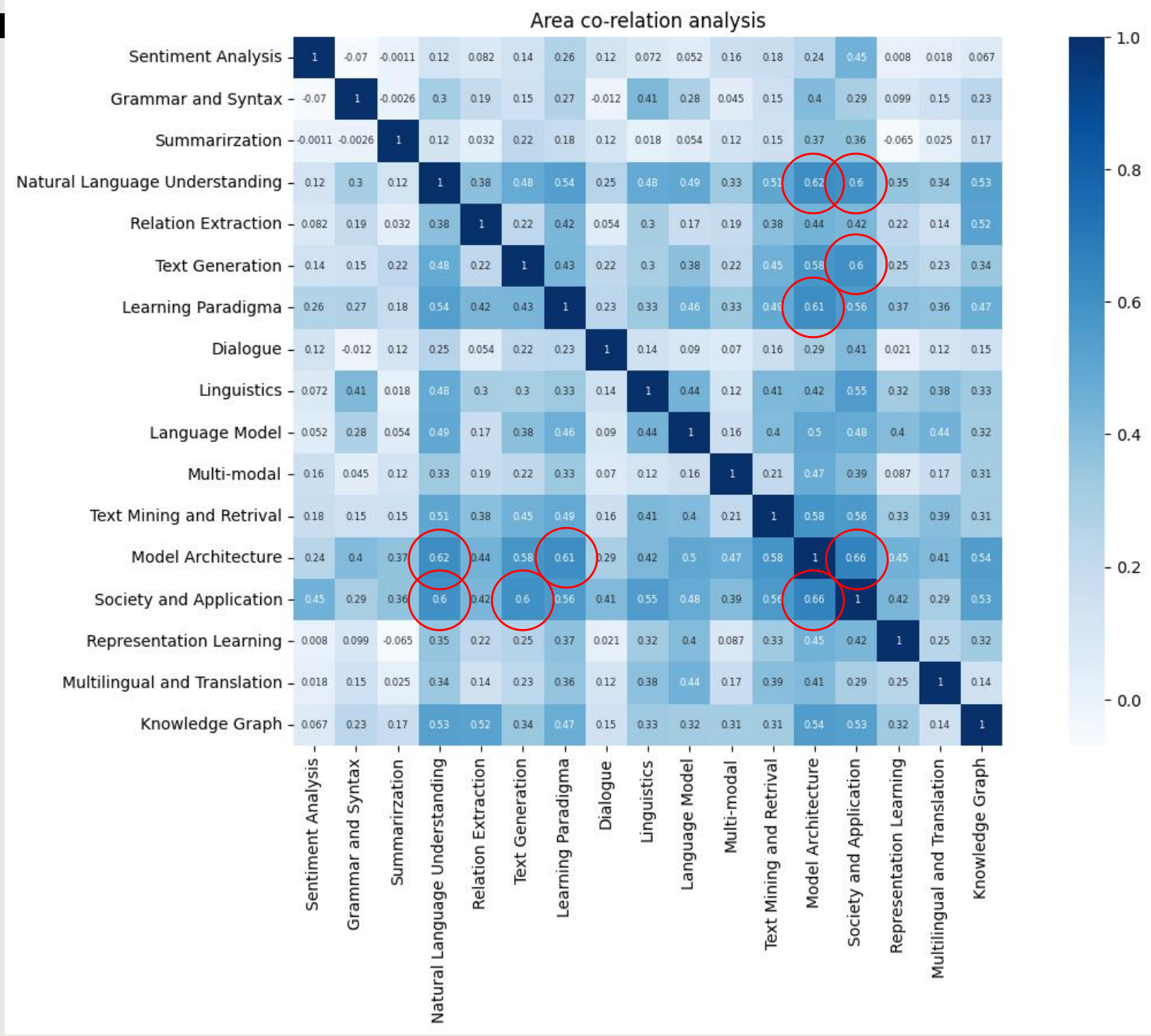
```
select p1.a, p2.a, count(*) from
    (select title, unnest(area) as a from paperInfo) as p1 join
    (select title, unnest(area) as a from paperInfo) as p2
    on p1.title = p2.title
group by p1.a, p2.a;
```

```
select p1.a, p2.a, count(*) from
    (select title, aId, unnest(area) as a from
        (select title, area, unnest(authorsId) as aid from paperInfo) as temp1) as p1 join
    (select title, aId, unnest(area) as a from
        (select title, area, unnest(authorsId) as aid from paperInfo) as temp2) as p2
    on p1.aId = p2.aId
group by p1.a, p2.a;
```

- Paper: areas of the same paper have high correlation



- Author: areas of the same author have high correlation



Potential Author Analysis

- Analyze potential and promising researchers in the NLP field
  - They don't have to be famous scholars

    paperCount < 20 & citationCount<300

  - Metircs
    - Paper count of their publication

```
select authorId, name, count(*) as paperCnt from
    (select * from authorInfo where paperCount < 20 and citationCount < 300) as newAuthor join
    paperInfo on newAuthor.authorId = any(paperInfo.authorsId)
group by authorId, name
order by paperCnt desc
limit 4;
```

    - Citation count of their publication

```
select authorId, name, sum(paperInfo.citationCount) as citationCnt from
    (select * from authorInfo where paperCount < 20 and citationCount < 300) as newAuthor join
    paperInfo on newAuthor.authorId = any(paperInfo.authorsId)
group by authorId, name
order by citationCnt desc
limit 4;
```

Potential Author Analysis

- Paper count of their publication

	authorId	name	paperCount
0	2065965333	Ivan Titov	6
1	72436283	Li Zhang	5
2	1830448175	Hongshen Chen	5
3	1845230025	Sudha Rao	5

- Citation count of their publication

	authorId	name	ciatationCount
0	65826567	Martin Josifoski	224
1	2111070044	Yuchen Ding	213
2	2145734278	Xin Zhao	213
3	152859769	Goutham Ramakrishnan	206

Potential Direction Analysis

- Analyze potential and emerging research direction in the NLP field

  Metric: gap of the keyword frequency

```
select keyword, max(keywordCnt::double precision/totalCnt::double precision) - min(keywordCnt::double precision/totalCnt::double precision) as frequencyGap
    from (select year, q, (year, q) as time, unnest(keywords) as keyword, count(*) as keywordCnt from
        (select *, ceil((split_part(text(publicationDate),'-',2))::numeric/3) as q from paperInfo) as temp1
        group by year, q, keyword)
    as keywordExtracted join
    (select (year, q) as time, count(*) as totalCnt from
        (select *, ceil((split_part(text(publicationDate),'-',2))::numeric/3) as q, unnest(keywords) as keyword from paperInfo) as temp2
        group by year, q
    ) as keywordTotal on keywordExtracted.time = keywordTotal.time
where keywordExtracted.year >= 2020
group by keyword
order by frequencyGap desc
limit 10
```

Potential Direction Analysis

	keyword	frequencyGap
0	natural language inference	0.104394
1	efficient federated learning framework	0.103524
2	neural machine translation	0.100146
3	question answering	0.059205
4	translation performance	0.052122
5	entity alignment	0.052122
6	translation quality	0.051969
7	entity recognition	0.050439
8	sequence pretraining	0.049854
9	contextual embeddings	0.042982

Summary

- Data collection –– Data Extraction — Question orientated analysis
- Focus:

1. NLP research areas:

   Author influences, Citation influences, Time trends, Inner correlation, Frequency changes (in keywords)

2. NLP popular authors:

   Beyond existed factors: cooperation frequency, predictions based on years data

- To Continue: More years, more properties, more Data Science

Thanks for Listening!

Also, our most sincere gratitude

to ***Prof. Gupta***

TA Ms. Kalyani Bhade

Dawei Li, Ruihan Wang, Jinhe Wen

Dec 8, 2022