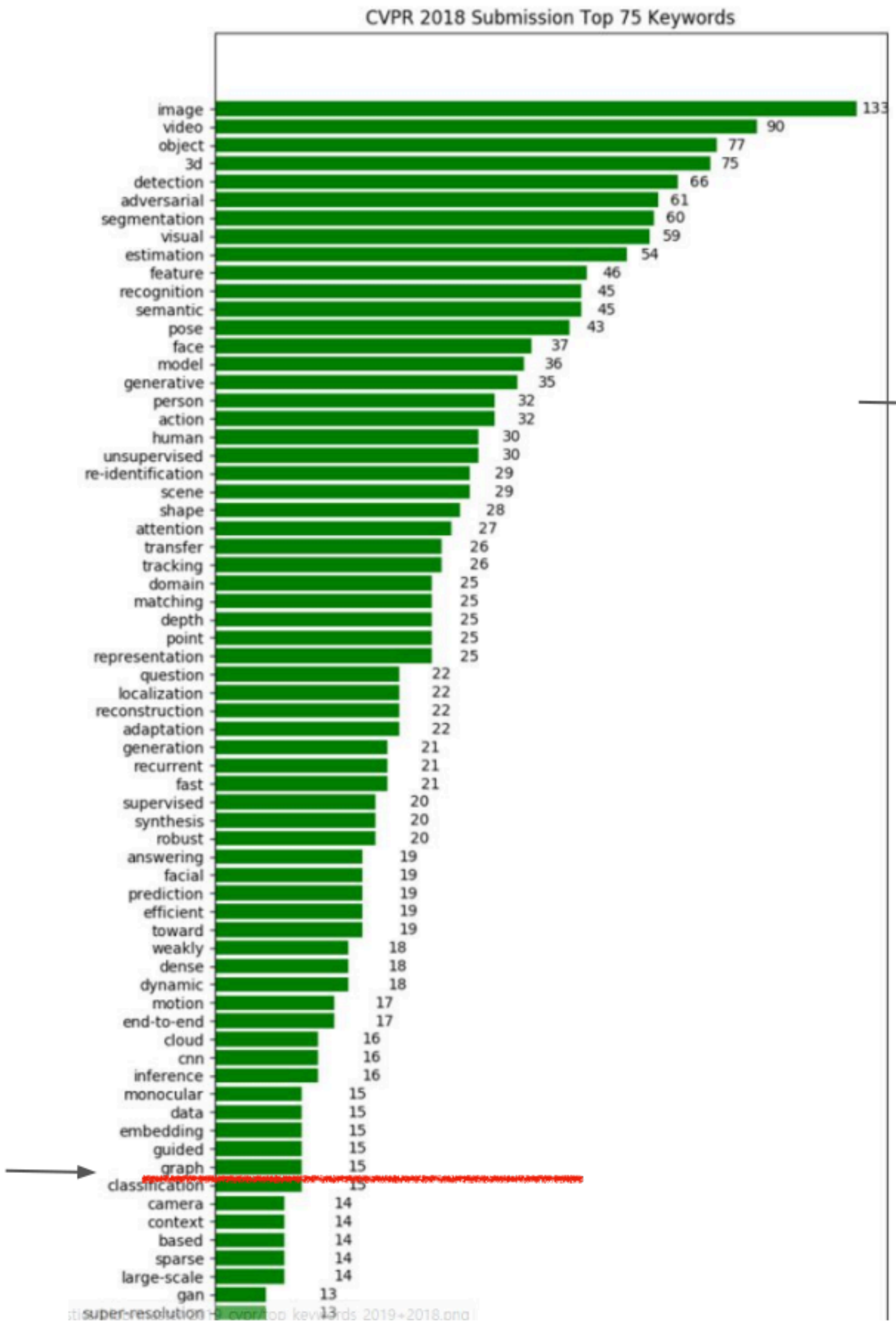
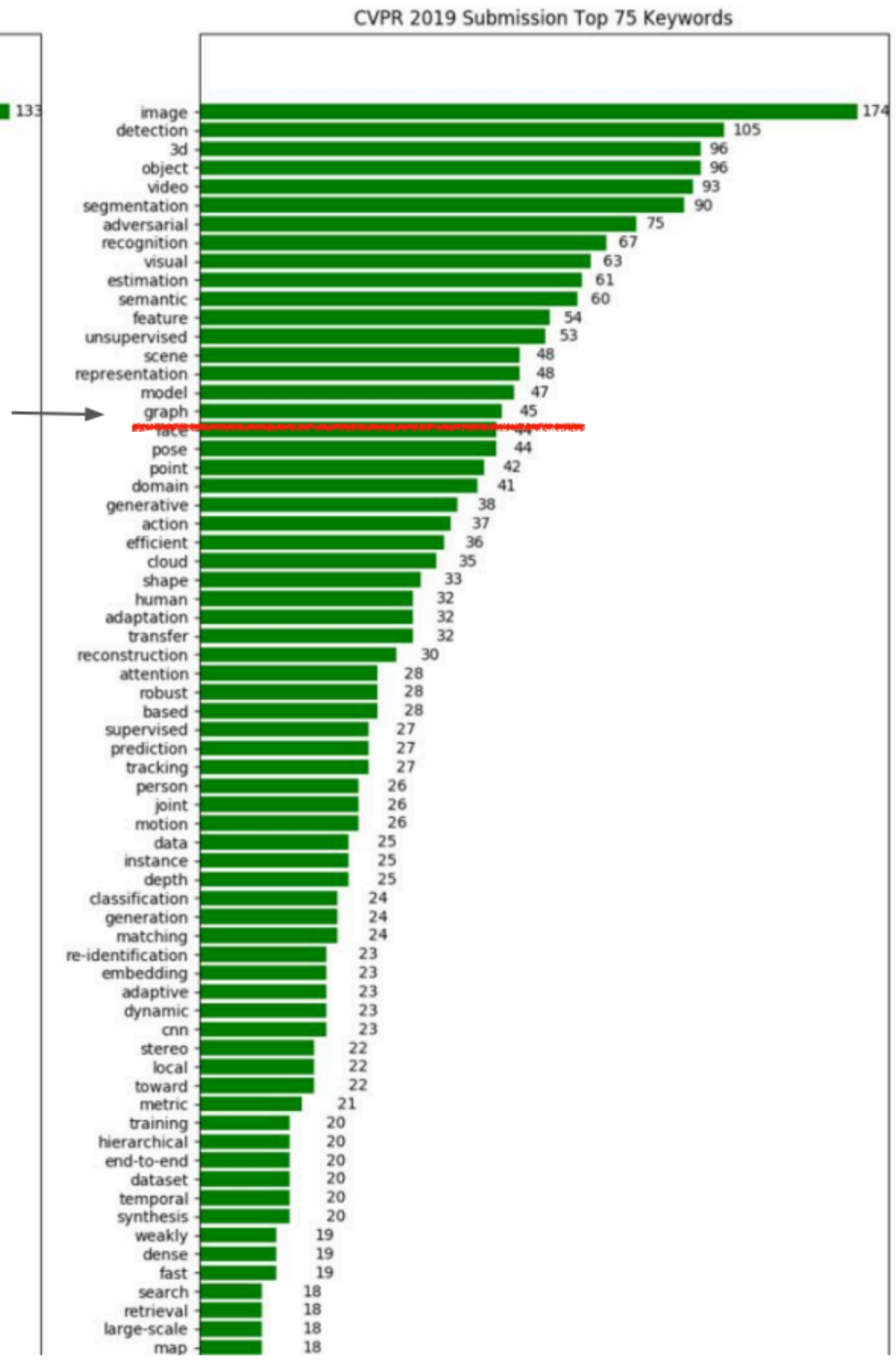
# Graph Convolutional Network

**Jaeyoung Cheong**

# 1. Trend

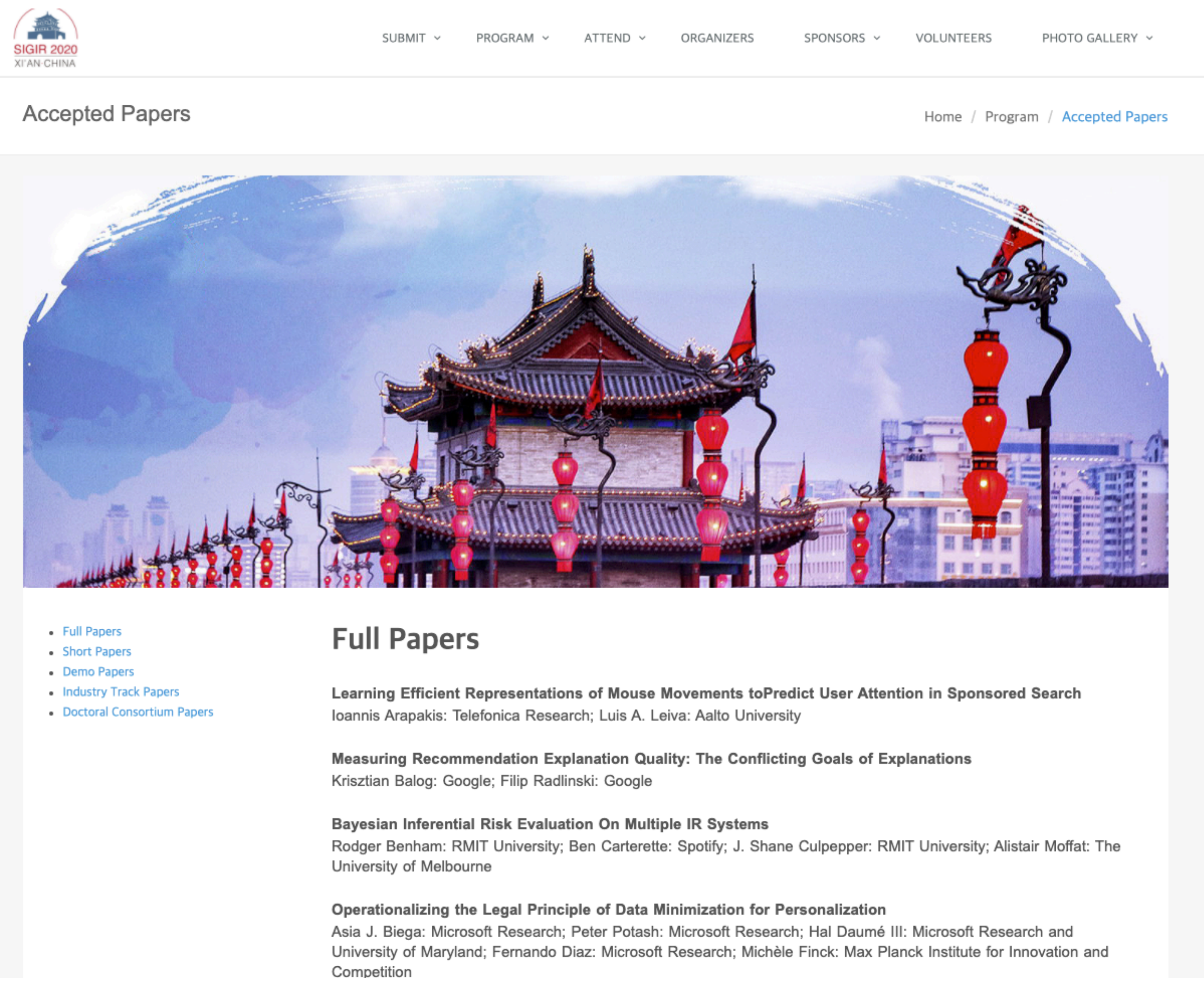
2019 CVPR



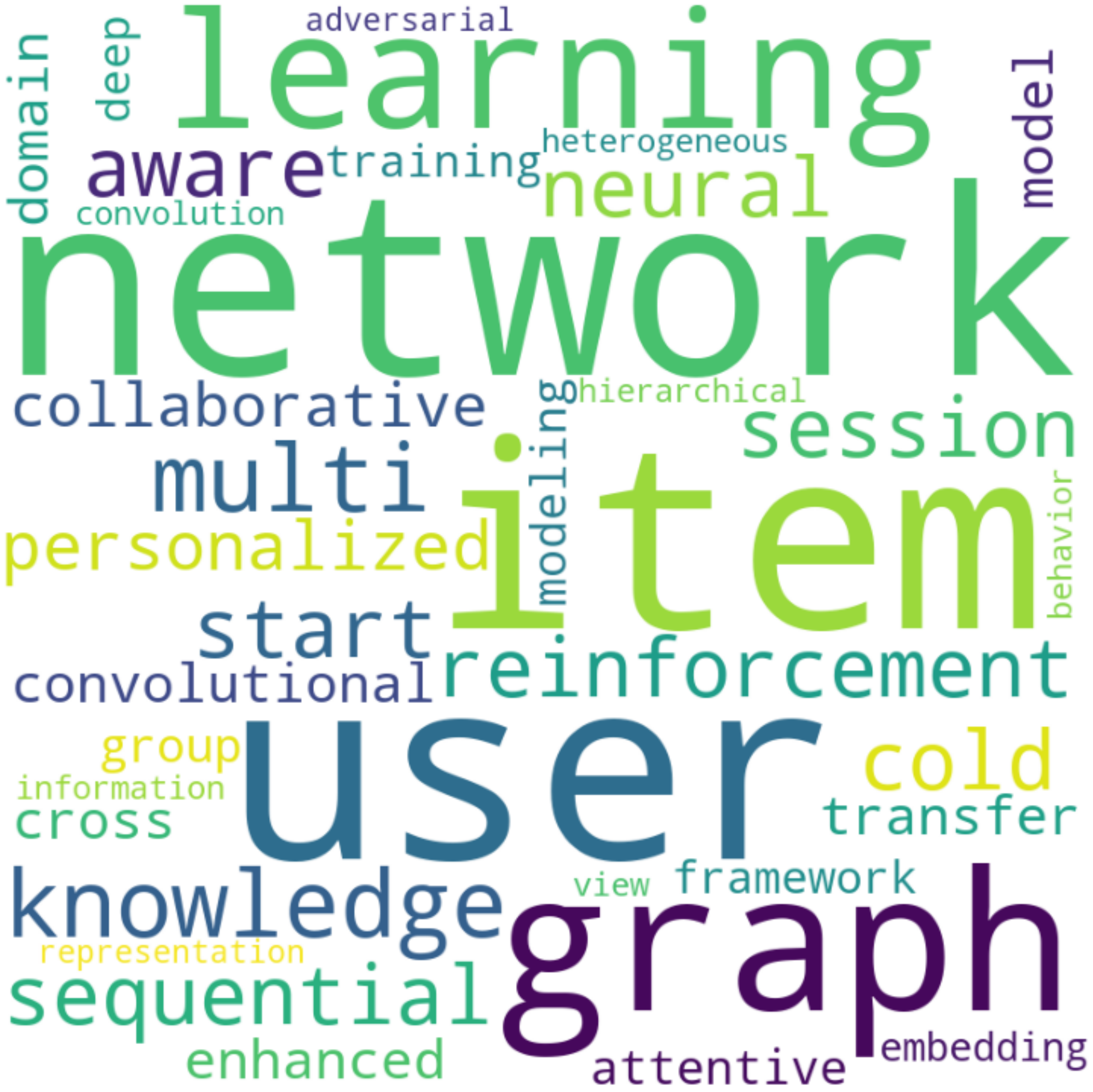
# 1. Trend

SIGIR 2020 ACCEPTED PAPERS




Ctrl+F: 'recommend'

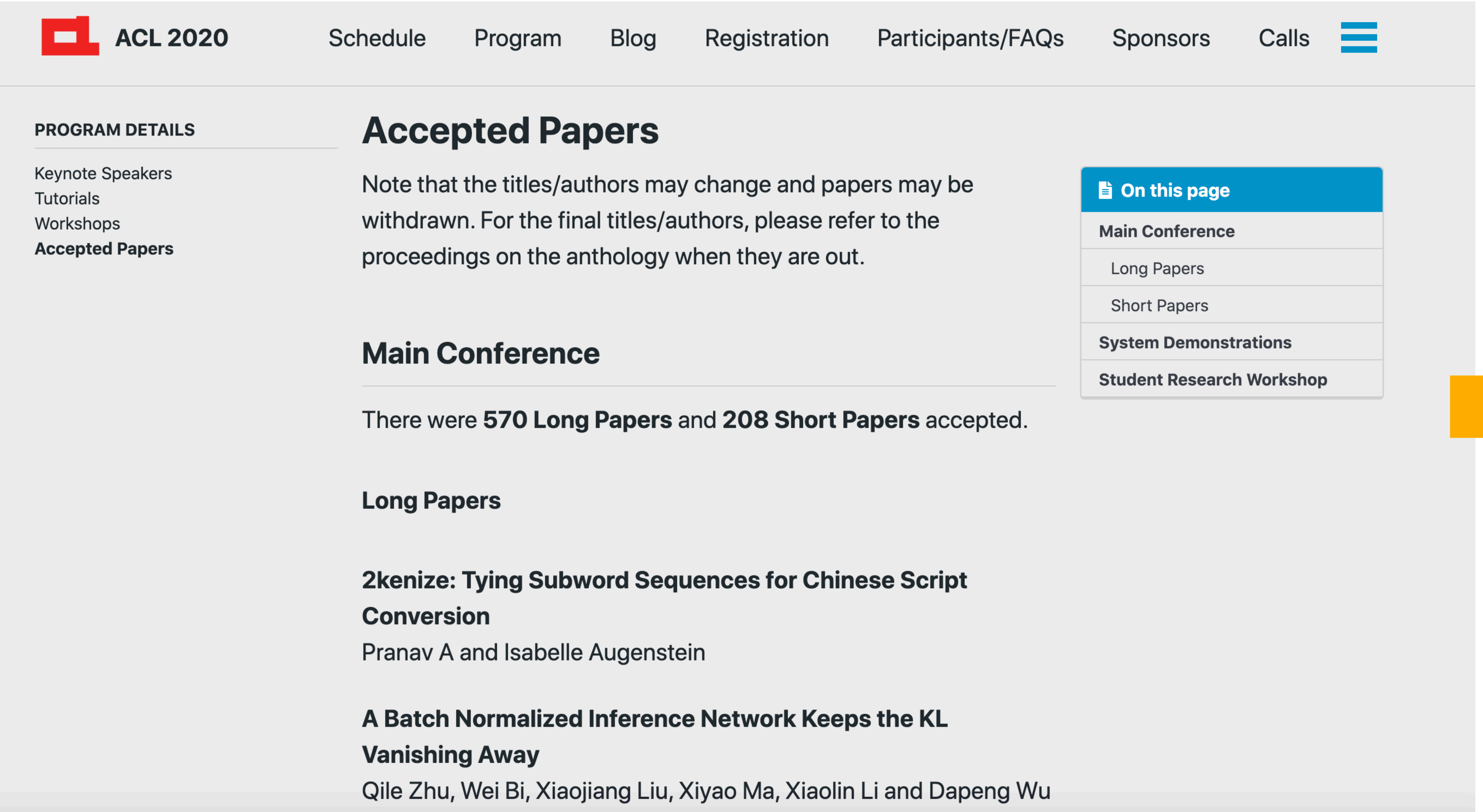
| | page | count |
|---|---|---|
| 0 | graph | 21 |
| 1 | learning | 18 |
| 2 | network | 18 |
| 3 | knowledge | 11 |
| 4 | multi | 9 |
| 5 | item | 9 |
| 6 | sequential | 9 |
| 7 | cold | 8 |
| 8 | start | 8 |
| 9 | aware | 7 |
| 10 | neural | 7 |
| 11 | session | 7 |
| 12 | reinforcement | 7 |
| 13 | personalized | 6 |
| 14 | user | 6 |
| 15 | users | 6 |
| 16 | collaborative | 5 |
| 17 | convolutional | 5 |
| 18 | transfer | 5 |
| 19 | cross | 5 |
| 20 | domain | 5 |
| 21 | enhanced | 4 |
| 22 | model | 4 |
| 23 | networks | 4 |
| 24 | attentive | 4 |




'recommendation' 같이 무정보적인 것은 불용어 처리함

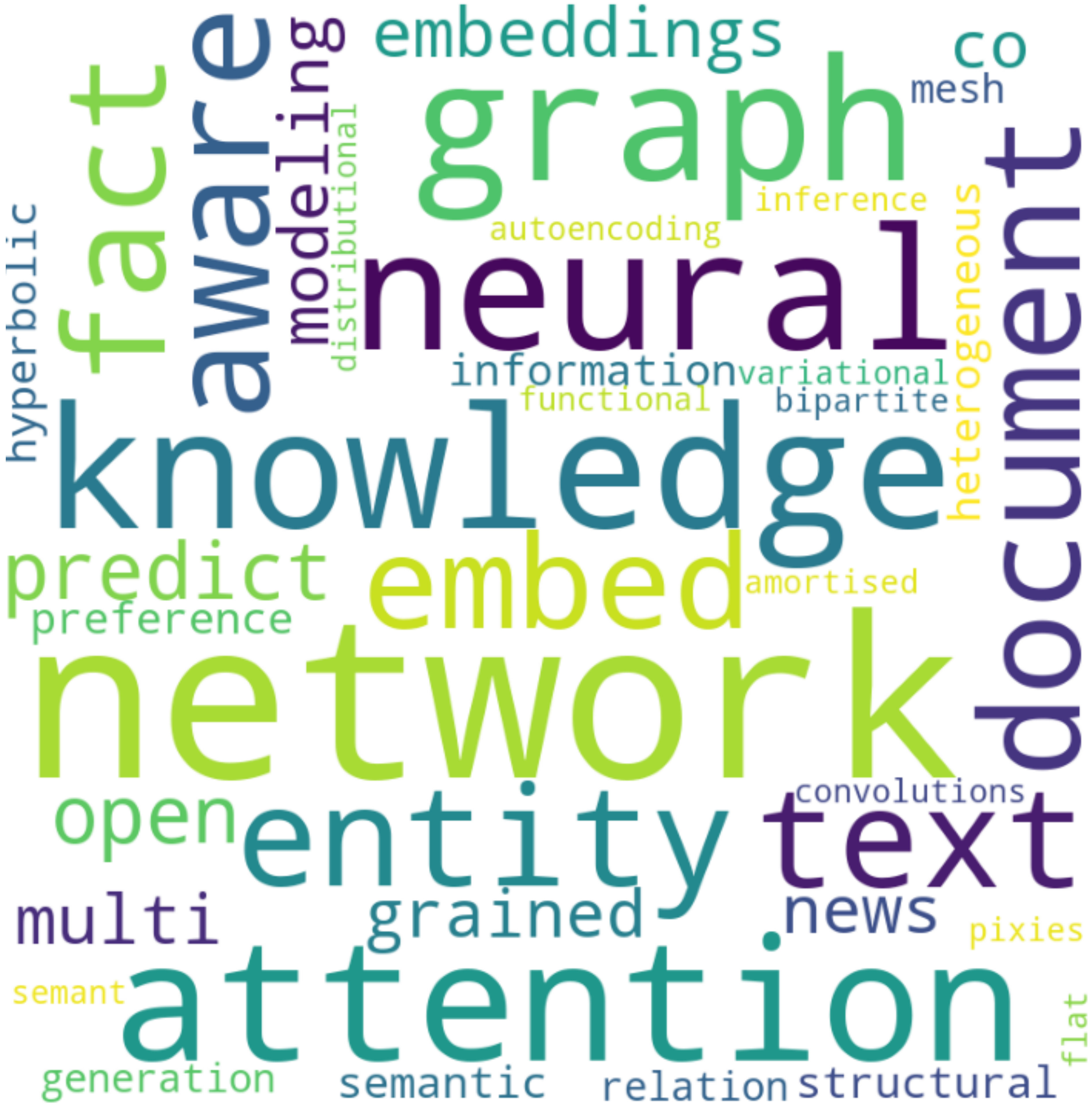
# 1. Trend

ACL 2020 ACCEPTED PAPERS




Ctrl+F: 'graph'

| | page | count |
|---|---|---|
| 0 | graph | 30 |
| 1 | network | 10 |
| 2 | knowledge | 7 |
| 3 | attention | 6 |
| 4 | networks | 5 |
| 5 | neural | 5 |
| 6 | entity | 3 |
| 7 | document | 3 |
| 8 | fact | 3 |
| 9 | aware | 3 |
| 10 | text | 3 |
| 11 | embed | 3 |
| 12 | predict | 2 |
| 13 | open | 2 |
| 14 | embeddings | 2 |
| 15 | modeling | 2 |
| 16 | multi | 2 |
| 17 | grained | 2 |
| 18 | co | 2 |
| 19 | news | 2 |
| 20 | preference | 2 |



# 1. Trend

ACL & SIGIR 정리 2020 accepted papers

| | page | count |
|---|---|---|
| 0 | graph | 21 |
| 1 | learning | 18 |
| 2 | network | 18 |
| 3 | knowledge | 11 |
| 4 | multi | 9 |
| 5 | item | 9 |
| 6 | sequential | 9 |
| 7 | cold | 8 |
| 8 | start | 8 |
| 9 | aware | 7 |
| 10 | neural | 7 |
| 11 | session | 7 |
| 12 | reinforcement | 7 |
| 13 | personalized | 6 |
| 14 | user | 6 |
| 15 | users | 6 |
| 16 | collaborative | 5 |
| 17 | convolutional | 5 |
| 18 | transfer | 5 |
| 19 | cross | 5 |
| 20 | domain | 5 |
| 21 | enhanced | 4 |
| 22 | model | 4 |
| 23 | networks | 4 |
| 24 | attentive | 4 |

## SIGIR

### Recommennder System

- Graph는 가장 HOT한 키워드
- 강화학습도 HOT
- Cold Start
- 개인화
- 협업 필터링
- 세션 기반 추천
- 신경망 기반 추천시스템

| | page | count |
|---|---|---|
| 0 | graph | 30 |
| 1 | network | 10 |
| 2 | knowledge | 7 |
| 3 | attention | 6 |
| 4 | networks | 5 |
| 5 | neural | 5 |
| 6 | entity | 3 |
| 7 | document | 3 |
| 8 | fact | 3 |
| 9 | aware | 3 |
| 10 | text | 3 |
| 11 | embed | 3 |
| 12 | predict | 2 |
| 13 | open | 2 |
| 14 | embeddings | 2 |
| 15 | modeling | 2 |
| 16 | multi | 2 |
| 17 | grained | 2 |
| 18 | co | 2 |
| 19 | news | 2 |
| 20 | preference | 2 |

## ACL

### Graph

- GNN으로 NLP 다룸
- Graph Attention Net의 급부상

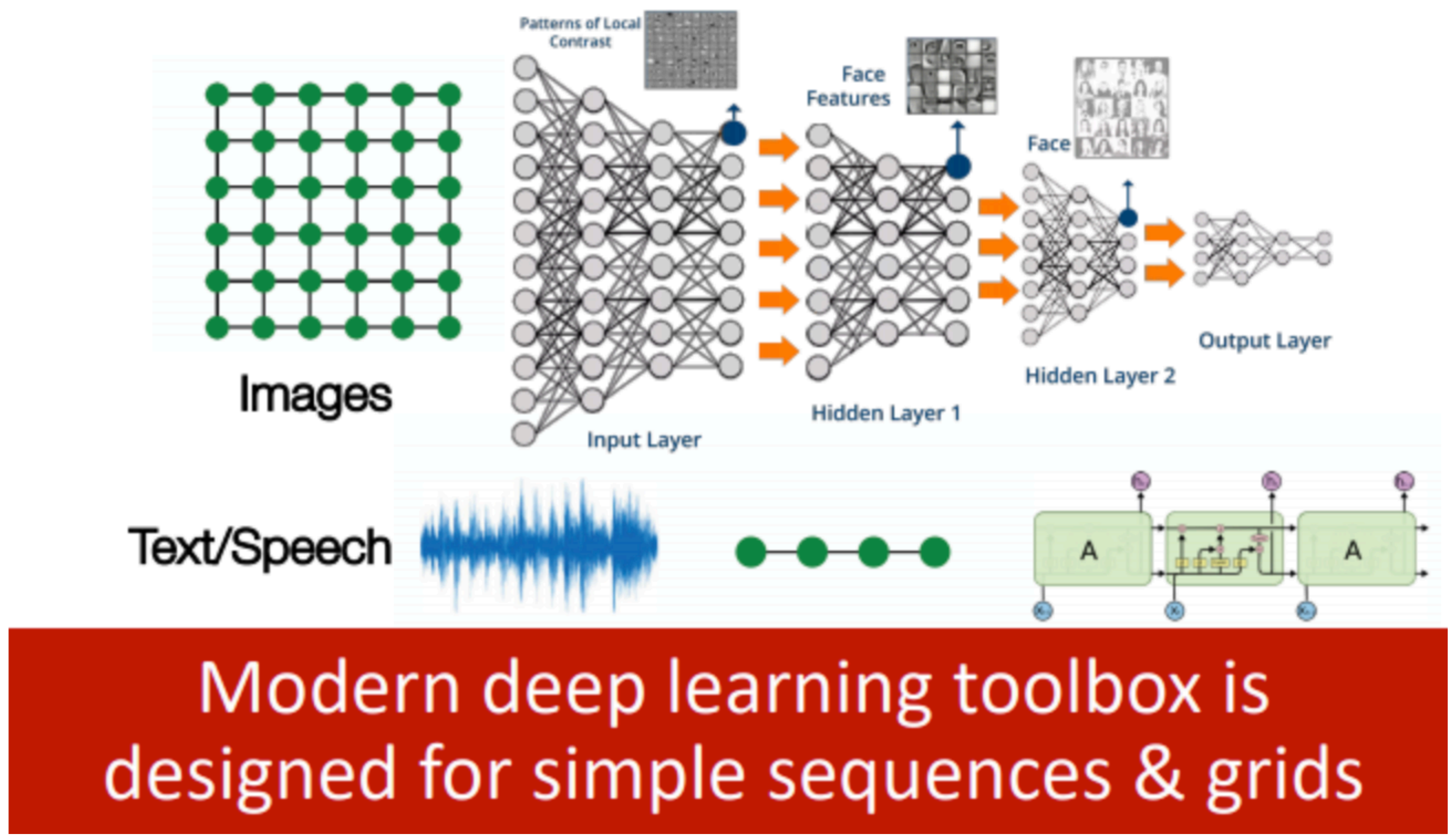
'recommendation' 같이 무정보적인 것은 불용어 처리함

# 2. Motivation

- 대부분의 성공적인 딥 or 머신러닝 작업에서 사용되던 데이터는 **유클리디안 공간(grid 위에서 표현된 공간)**에서 표현되어진다.

- 하지만 데이터가 비유클리디안 공간에서 만들어지거나 **복잡한 관계를 가진 그래프**로 표현된 경우,
기존의 머신러닝 알고리즘으로 이것을 처리하기 어렵다. (그래프는 비유클리드 공간 위에서 표현됨)

- e-커머스: 구매자와 상품 간의 복잡한 상호관계를 다룬다.
- 화학: 분자들은 그래프로 모델화된다.




**ConvGNNs**

컴퓨터 비전에서 CNN의 성공이 동기가 되어, 그래프 데이터의. Convolution 개념에 대한 재정의가 이루어졌다.

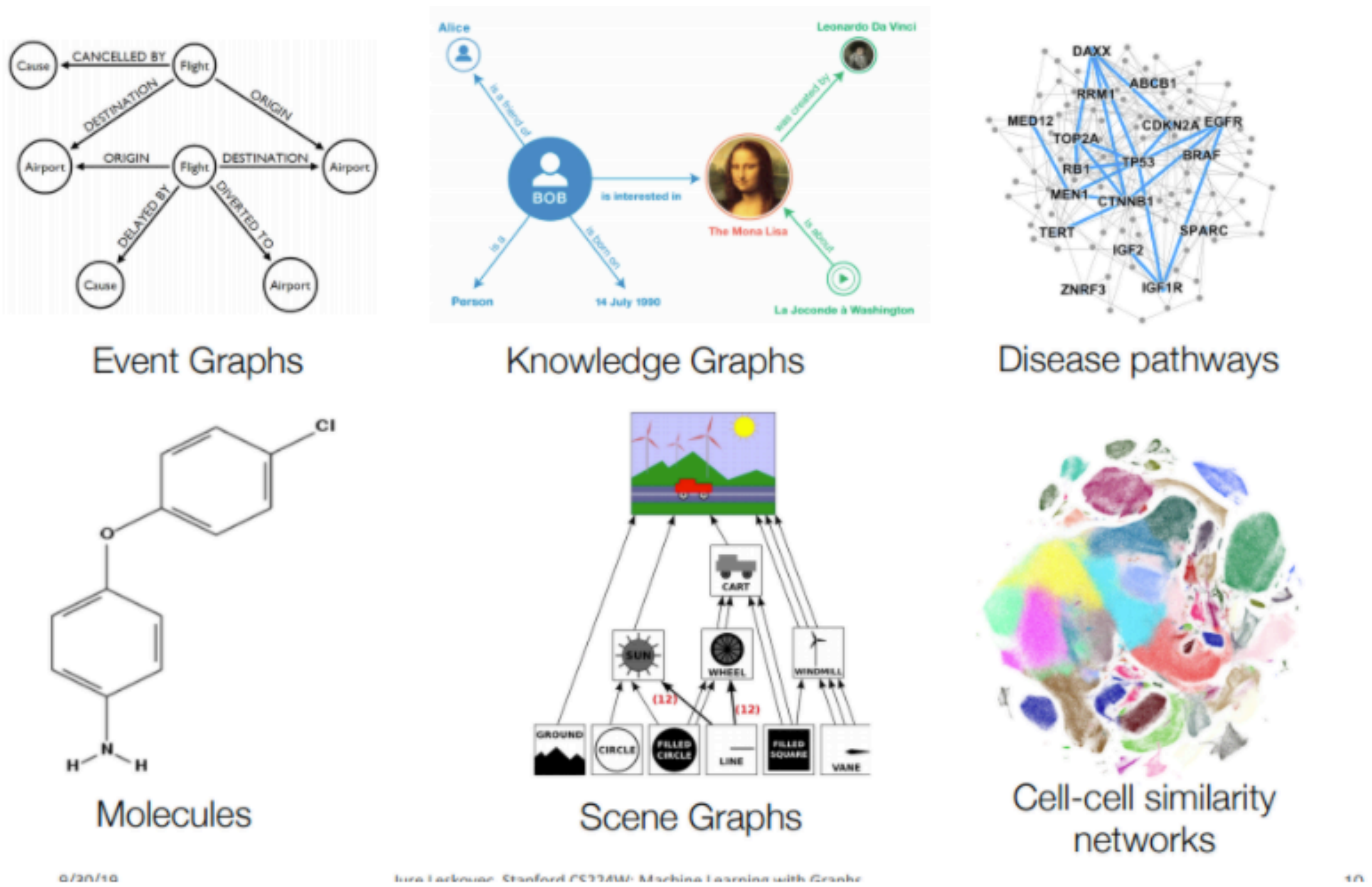
- the spectral-based approaches (Bruna et al 2013)
- The spatial-based approaches

**Reccurent GNNs**

**AutoEncoder GNNs**

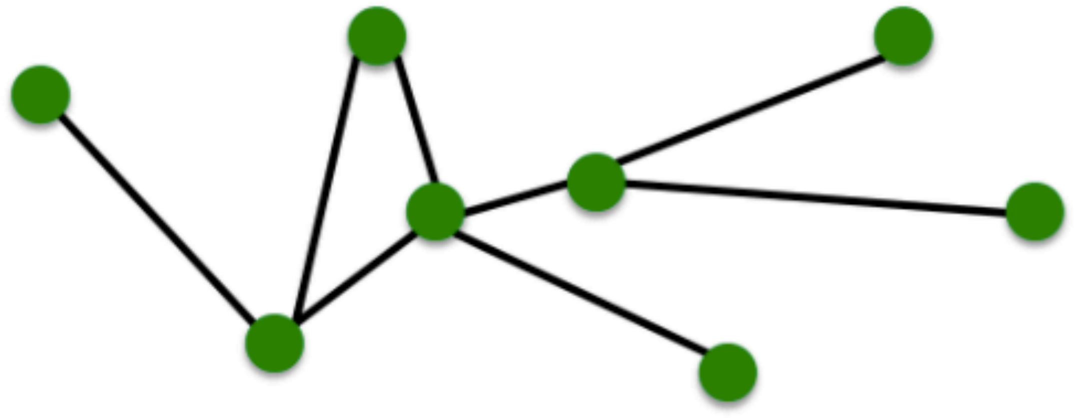
**Graph Attention Networks**

**...**

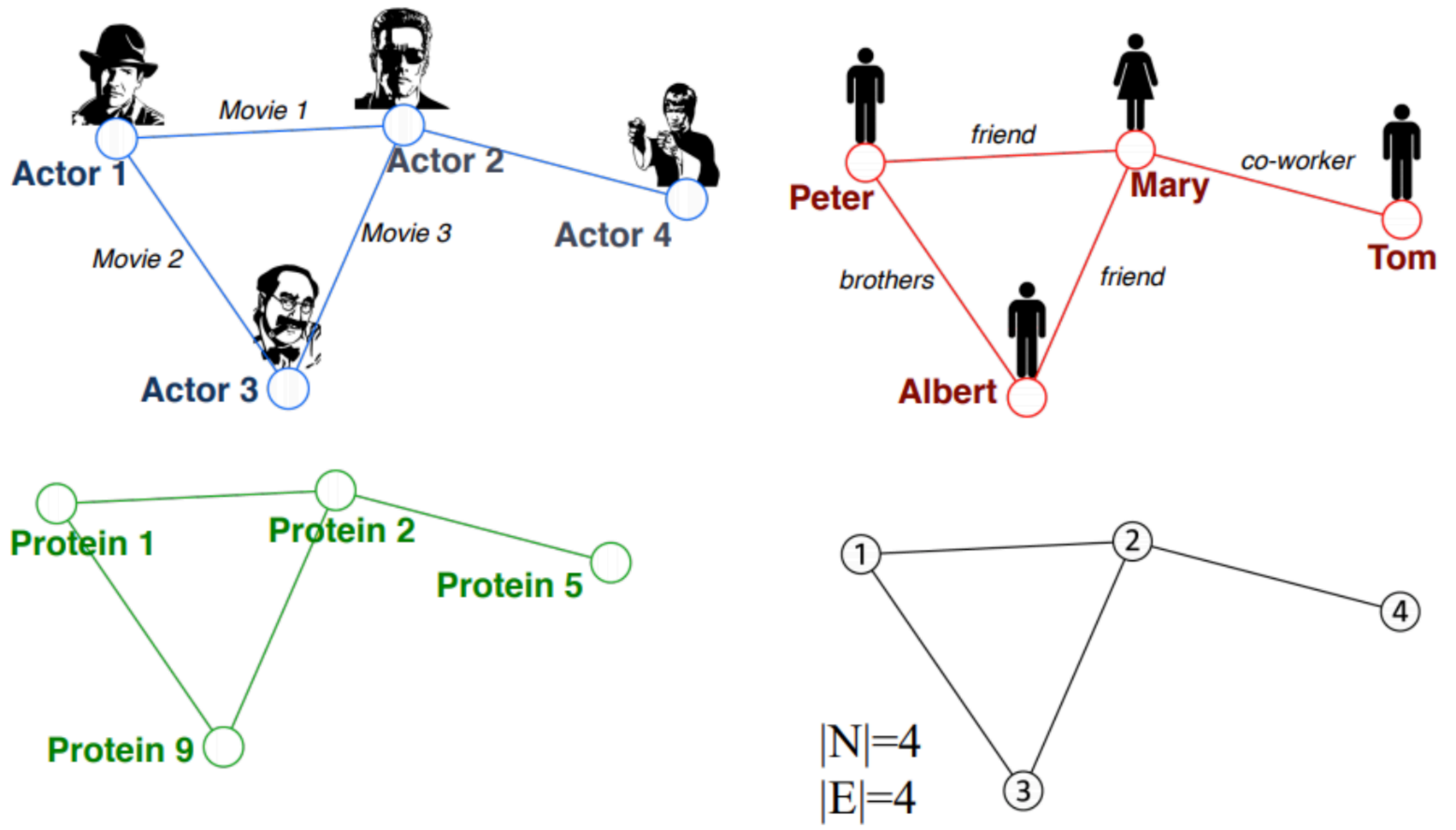


# 3. Graph

## 3.1 Def

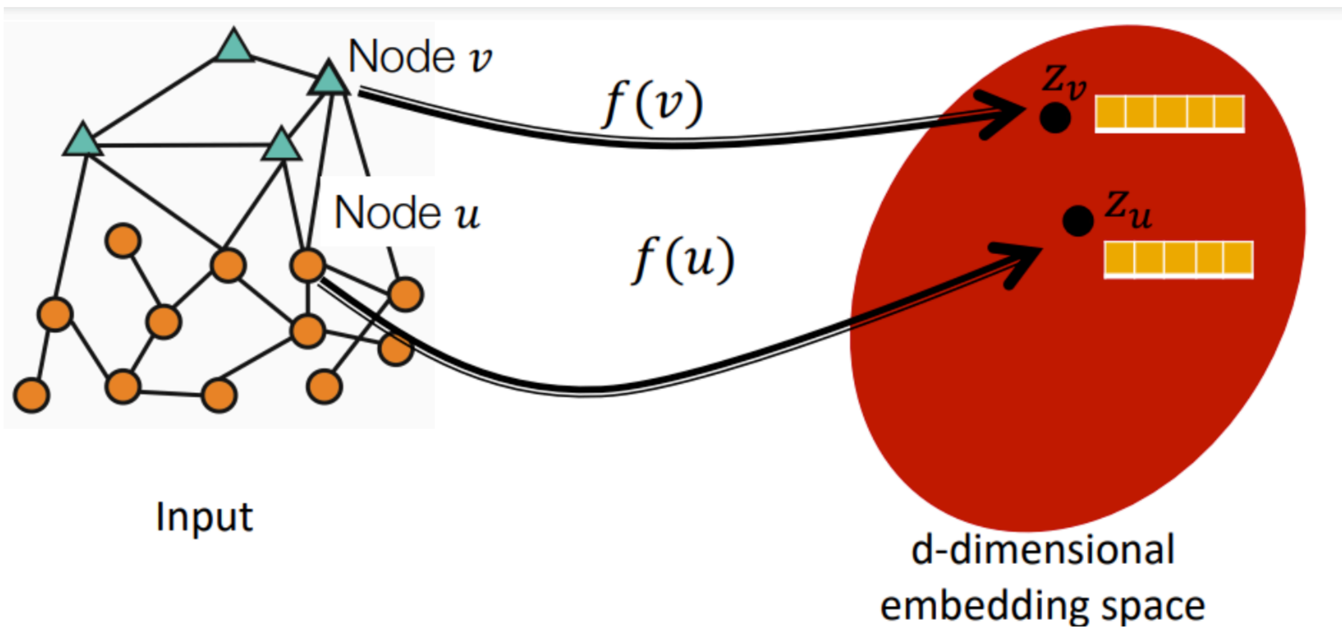


- **Objects:** nodes, vertices — ***N***
- **Interactions:** links, edges — ***E***
- **System:** network, graph — ***G(N,E)***



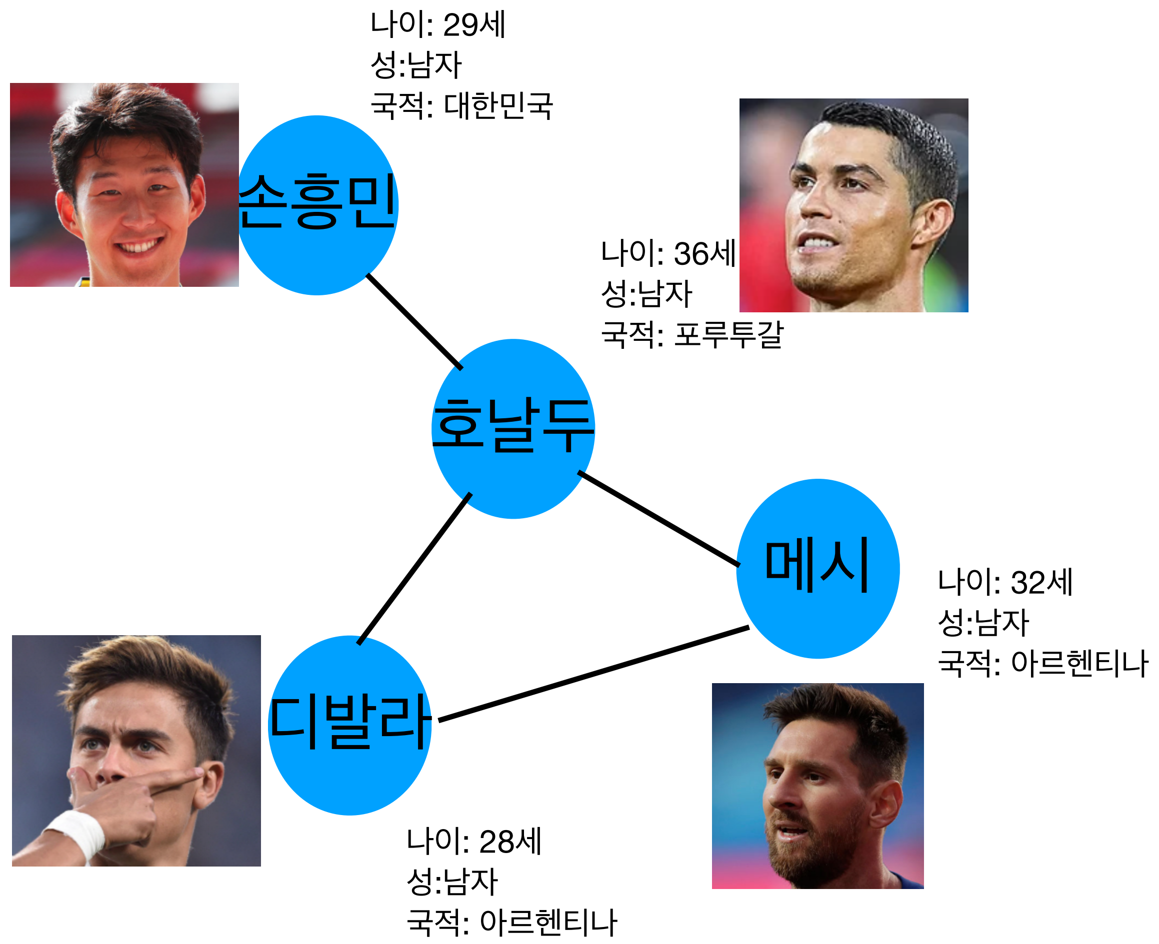

## 3.2 Embedding Node

그래프의 node를 d 차원으로 임베딩. 여기서 유사한 이웃을 가진 노드들은 서로 가까이 있게끔 임베딩된다.



# 3. Graph

## 3.3 How to represent graph into the matrix




| | 국적 | Feature2 | Feature3 | |
|---|---|---|---|---|
| son | 0 | 1 | 0 | 대한민국 |
| Ronaldo | 1 | 1 | 1 | 포르투갈 |
| Messi | 1 | 1 | 1 | 둘다 아르헨티나 |
| Dybala | 1 | 1 | 1 | |

국적을 나타내는 Features라고 해보자

N(node) x F(features)의 **Node Feature Matrix** $X_{ij}$

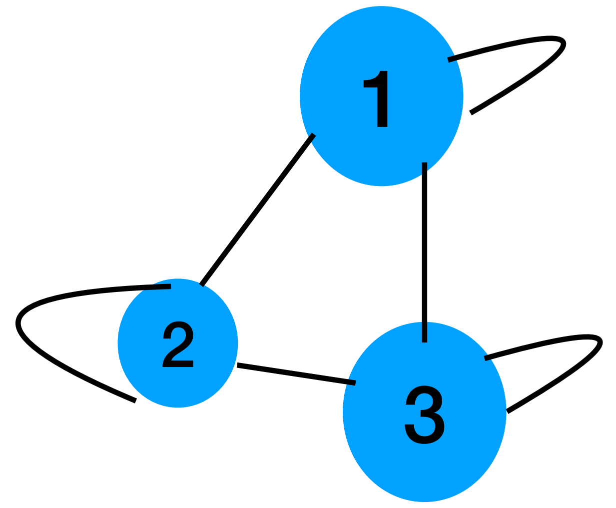
| | son | Ronaldo | Messi | Dybala |
|---|---|---|---|---|
| son | 0 | 1 | 0 | 0 |
| Ronaldo | 1 | 0 | 1 | 1 |
| Messi | 0 | 1 | 0 | 1 |
| Dybala | 0 | 1 | 1 | 0 |

N x N(node)의 **Adjacent Matrix** $A_{ij}$

| | son | Ronaldo | Messi | Dybala |
|---|---|---|---|---|
| son | 1 | 0 | 0 | 0 |
| Ronaldo | 0 | 3 | 0 | 0 |
| Messi | 0 | 0 | 2 | 0 |
| Dybala | 0 | 0 | 0 | 2 |

$D$ Degree matrix

# 4. Graph Convolutional Networks




## 4.1 The limits of adjacent matrix

### 1) 각 노드 자신에 대한 정보가 학습 시 업데이트되지 않는다.

→ 각 Mat에 I를 더해줌으로써, 각 노드 자기 자신에 대한 정보도 학습 시 업데이트할 수 있다.

| | son | Ronaldo | Messi | Dybala |
|---|---|---|---|---|
| son | 1 | 0 | 0 | 0 |
| Ronaldo | 0 | 3 | 0 | 0 |
| Messi | 0 | 0 | 2 | 0 |
| Dybala | 0 | 0 | 0 | 2 |

$D$ Degree matrix

+

| | son | Ronaldo | Messi | Dybala |
|---|---|---|---|---|
| son | 1 | 0 | 0 | 0 |
| Ronaldo | 0 | 1 | 0 | 0 |
| Messi | 0 | 0 | 1 | 0 |
| Dybala | 0 | 0 | 0 | 1 |

I

=

| | | | |
|---|---|---|---|
| 2 | 0 | 0 | 0 |
| 0 | 4 | 0 | 0 |
| 0 | 0 | 3 | 0 |
| 0 | 0 | 0 | 3 |

$\widetilde{D}$

| | son | Ronaldo | Messi | Dybala |
|---|---|---|---|---|
| son | 0 | 1 | 0 | 0 |
| Ronaldo | 1 | 0 | 1 | 1 |
| Messi | 0 | 1 | 0 | 1 |
| Dybala | 0 | 1 | 1 | 0 |

$A_{ij}$ Adjacent Matrix

+

| | son | Ronaldo | Messi | Dybala |
|---|---|---|---|---|
| son | 1 | 0 | 0 | 0 |
| Ronaldo | 0 | 1 | 0 | 0 |
| Messi | 0 | 0 | 1 | 0 |
| Dybala | 0 | 0 | 0 | 1 |

I

=

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |

$\widetilde{A}_{ij}$

# 4. Graph Convolutional Networks

## 4.1 The limits of adjacent matrix

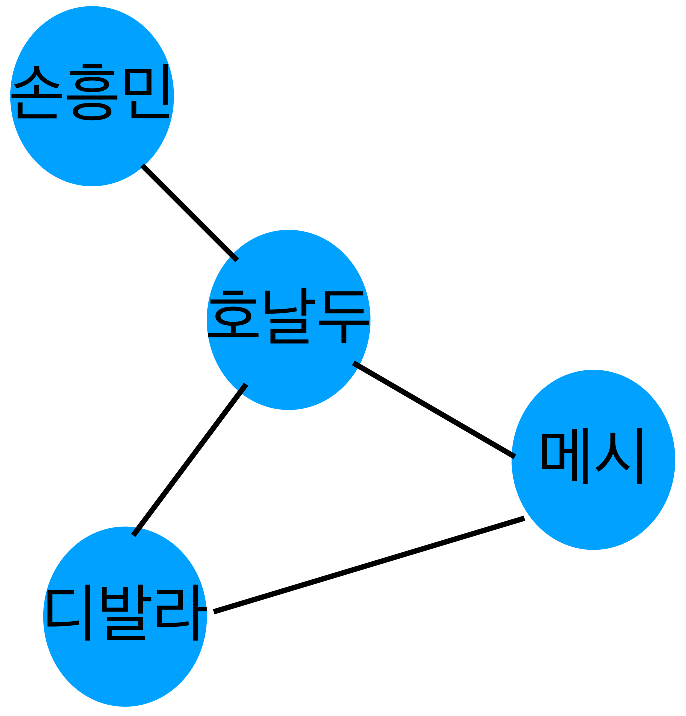
2) 기본적인 인접행렬 A는 정규화가 되어있지 않아 gradient explosion을 발생시키거나 feature matrix와 곱할 때(AX) X의 크기를 불안정하게 크게 만들 수 있다.

해소방안: $\widetilde{A} \longrightarrow \widehat{A} = \widetilde{D}^{-0.5}\,\widetilde{A}\,\widetilde{D}^{-0.5}$

# 4. Graph Convolutional Networks

## 4.1 The limits of adjacent matrix

2) 기본적인 인접행렬 A는 정규화가 되어있지 않아 gradinet explosion을 발생시키거나 feature matrix와 곱할 때(AX) X의 크기를 불안정하게 크게 만들 수 있다.




| | | | |
|---|---|---|---|
| 2 | 0 | 0 | 0 |
| 0 | 4 | 0 | 0 |
| 0 | 0 | 3 | 0 |
| 0 | 0 | 0 | 3 |

$\tilde{D}$

→

| | | | |
|---|---|---|---|
| 1/2 | 0 | 0 | 0 |
| 0 | 1/4 | 0 | 0 |
| 0 | 0 | 1/3 | 0 |
| 0 | 0 | 0 | 1/3 |

$\tilde{D}^{-1}$

평균으로 정규화

| | | | |
|---|---|---|---|
| 1/2 | 0 | 0 | 0 |
| 0 | 1/4 | 0 | 0 |
| 0 | 0 | 1/3 | 0 |
| 0 | 0 | 0 | 1/3 |

$\tilde{D}^{-1}$

*

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |

$\tilde{A}_{ij}$

→

| | | | |
|---|---|---|---|
| 1/2 | 1/2 | 0 | 0 |
| 1/4 | 1/4 | 1/4 | 1/4 |
| 0 | 1/3 | 1/3 | 1/3 |
| 0 | 1/3 | 1/3 | 1/3 |

$S$

고차 노드에 더 작은 수를 곱함으로써 영향력을 줄이기

*

| | | | |
|---|---|---|---|
| 1/2 | 0 | 0 | 0 |
| 0 | 1/4 | 0 | 0 |
| 0 | 0 | 1/3 | 0 |
| 0 | 0 | 0 | 1/3 |

$\tilde{D}^{-1}$

=

| | | | |
|---|---|---|---|
| 1/2 | 1/8 | 0 | 0 |
| 1/8 | 1/16 | 1/12 | 1/12 |
| 0 | 1/12 | 1/9 | 1/9 |
| 0 | 1/12 | 1/9 | 1/9 |

$\hat{A}$

# 4. Graph Convolutional Networks

## 4.1 The limits of adjacent matrix

| | | | |
|---|---|---|---|
| 2 | 0 | 0 | 0 |
| 0 | 4 | 0 | 0 |
| 0 | 0 | 3 | 0 |
| 0 | 0 | 0 | 3 |

$\tilde{D}$

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |

$\tilde{A}_{ij}$

$$\hat{A} = \tilde{D}^{-1} \tilde{A} \tilde{D}^{-1}$$

이 같은 계산은 사실상 정규화를 두 번 하는 것이기 때문에 아래 같이 수정함으로써 rebalance 해주기

$$\hat{A} = \tilde{D}^{-0.5} \tilde{A} \tilde{D}^{-0.5}$$

Normalized Adjacency Matrix $\longrightarrow$ $H^{(l+1)} = \sigma\left(\underbrace{\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}}_{= \hat{A}} H^{(l)} W^{(l)}\right).$

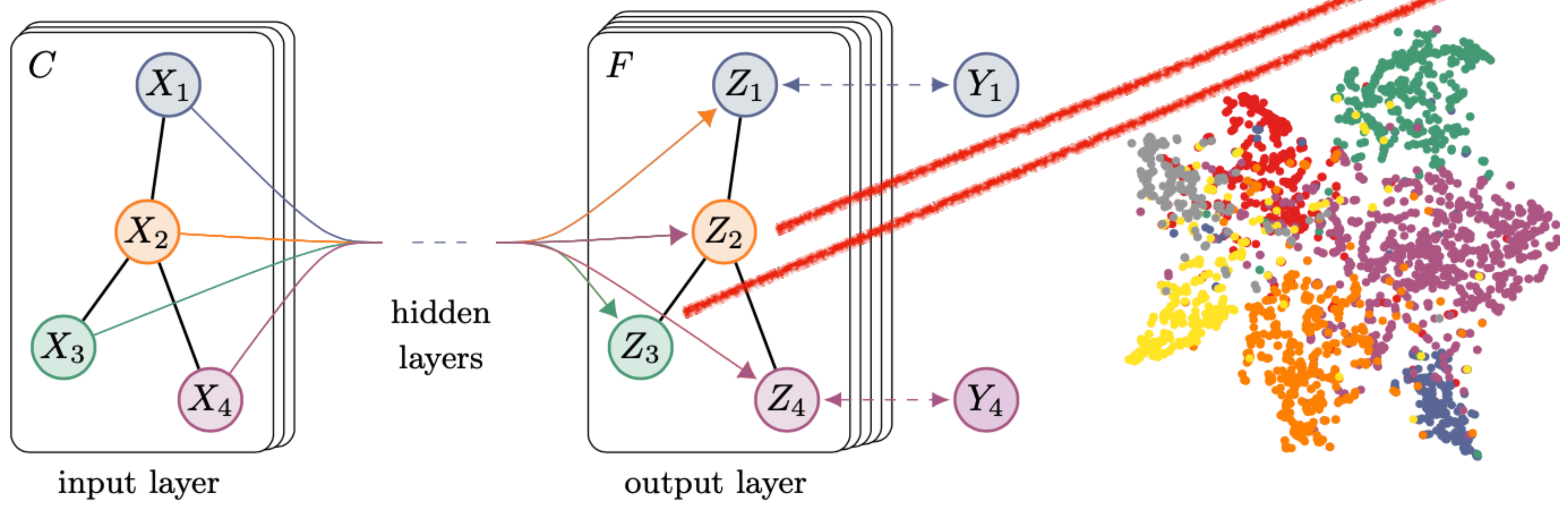
- 어떤 노드가 서로 연결되었는지
- 각 노드 간의 연결 강도가 얼마인지

- H: node feature matrix
- W: weight

# 4. Graph Convolutional Networks

we consider a two-layer GCN for semi-supervised node classification on a graph with a symmetric adjacency matrix A

라벨이 없는 것도 분류를 한다는 점에서 준지도학습을 잘 수행




(a) Graph Convolutional Network

(b) Hidden layer activations

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$$

$$Z = f(X, A) = \mathrm{softmax}\Big(\hat{A}\ \mathrm{ReLU}\Big(\hat{A} X W^{(0)}\Big) W^{(1)}\Big)$$

Layer1

Layer2

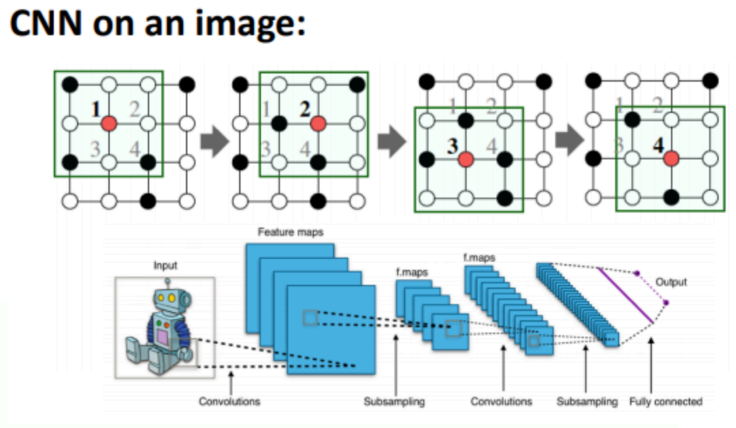
$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf} ,$$

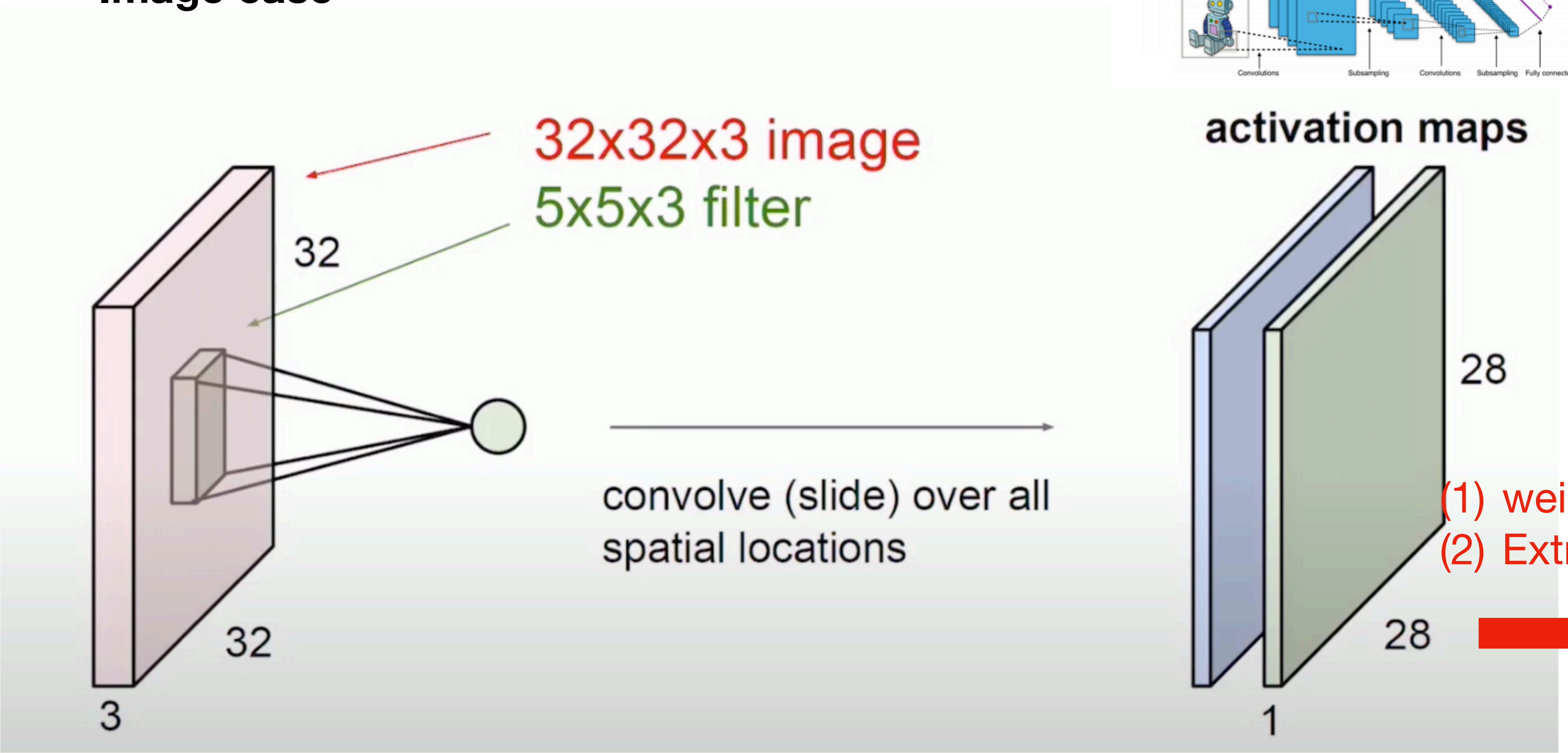
Y: 실제 값, Z: 예측 값: Y,Z의 크로스엔트로피

# 4. Graph Convolutional Networks

4.1 Convolution

**Image case**



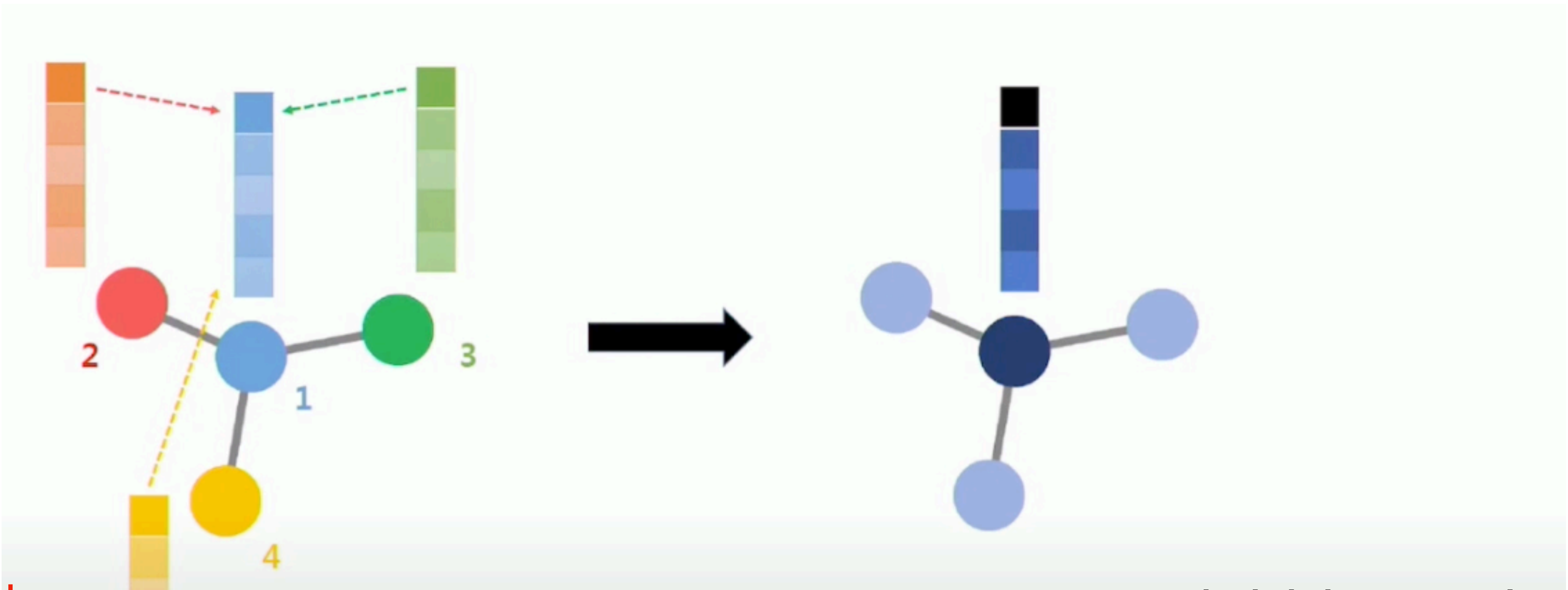




(1) weight sharing
(2) Extracting local features

- Filter: 한 층 당 25 개의 파라미터. 3 개의 층이니까 75 개 파라미터가 있다
- 저 Filter는 정해진 stride 만큼 건너 뛰면서 전체 image를 조사
- 몇 개의 필터를 사용했는가에 따라 activation map의 depth 결정된다

- **UPDATE:** Conv를 하나 거칠 때 마다 (**이미지**의 특성을 담고 있는) activation map의 값이 업데이트

**Graph case**




**1 st node와 연결된 node들만을 더해줌**

$$H_2^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + H_4^{(l)}W^{(l)} + b^{(l)}\right)$$

1

$$\Rightarrow \quad H_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)} H_j^{(l)}W^{(l)} + b^{(l)}\right)$$

- **W**: Weight
- **H**: node feature matrix
- $H_n$: n th node의 feature 값들
- $l$: l th layer

- **UPDATE:** Conv를 하나 거칠 때 마다 (**노드**의 특성을 담고 있는) activation map의 값이 업데이트

# 4. Graph Convolutional Networks

4.2 Compact Forms

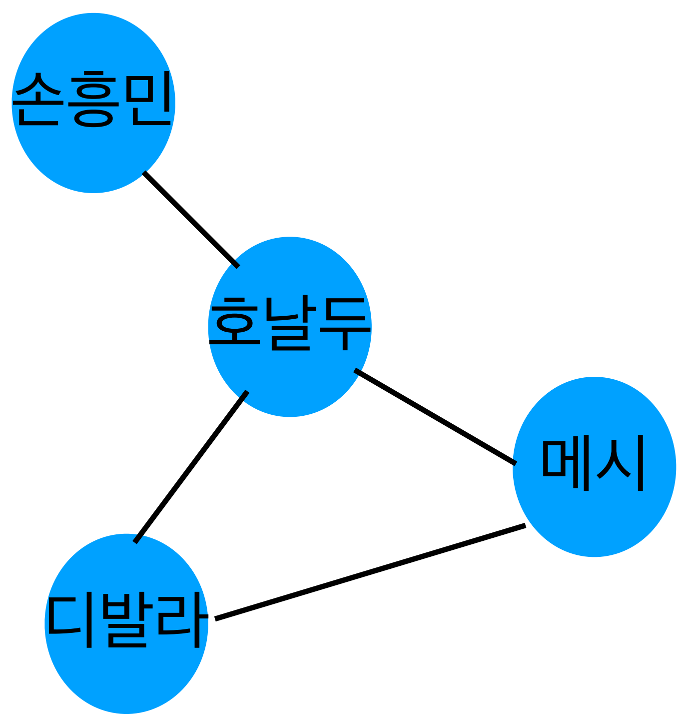
$$H_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} H_j^{(l)} W^{(l)} + b^{(l)}\right)$$

→

Adjacent Matrix

$$H^{(l+1)} = \sigma\left(A H^{(l)} W^{(l)} + b^{(l)}\right)$$

learnable parameters are shared




| | son | Ronaldo | Messi | Dybala |
|---|---|---|---|---|
| son | 1/2 | 1/8 | 0 | 0 |
| Ronaldo | 1/8 | 1/16 | 1/12 | 1/12 |
| Messi | 0 | 1/12 | 1/9 | 1/9 |
| Dybala | 0 | 1/12 | 1/9 | 1/9 |

Normalized
**Adjacent Matrix**
**(Ahat 4,4)**

| | feature 1 | Feature 2 | Feature 3 | feature 4 | Feature 5 |
|---|---|---|---|---|---|
| son | 1 | 1 | 1 | 0 | 1 |
| Ronaldo | 1 | 0 | 1 | 1 | 0 |
| Messi | 0 | 1 | 1 | 1 | 1 |
| Dybala | 0 | 1 | 1 | 1 | 1 |

**Node feature Matrix**
**(X 4,5)**

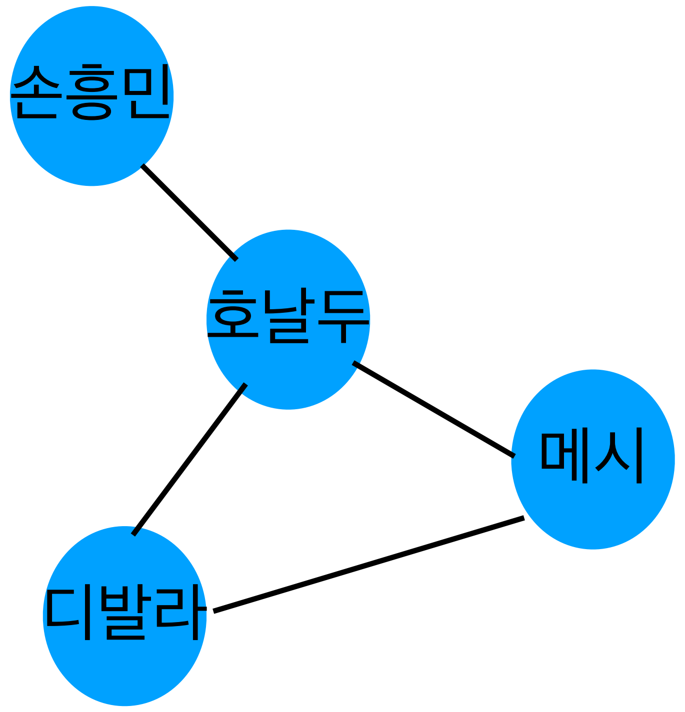
| | Filter1 | Filter2 | … | Filter64 |
|---|---|---|---|---|
| feature1 | 1 | 1 | 1 | 0 |
| Feature2 | 1 | 0 | 1 | 1 |
| Feature3 | 0 | 1 | 1 | 1 |
| Feature4 | 0 | 1 | 1 | 1 |
| Feature5 | 1 | 1 | 0 | 1 |

**Weight**
**(W 5,64)**

자기 자신과는 연결되어 있지는 않지만, 이전 레이어에서 다음 레이어로 갈 때, 다음 레이어에서 이전 것의 정보를 받아야 하니까 자기 자신은 1로 가정

# 4. Graph Convolutional Networks

4.2 Compact Forms




| | feature 1 | Feature 2 | Feature 3 | feature 4 | Feature 5 |
|---|---|---|---|---|---|
| son | 1 | 1 | 1 | 0 | 1 |
| Ronaldo | 1 | 0 | 1 | 1 | 0 |
| Messi | 0 | 1 | 1 | 1 | 1 |
| Dybala | 0 | 1 | 1 | 1 | 1 |

**Node feature Matrix (X 4,5)**

*

| | Filter1 | Filter2 | ... | Filter64 |
|---|---|---|---|---|
| feature1 | 1 | 1 | 1 | 0 |
| Feature2 | 1 | 0 | 1 | 1 |
| Feature3 | 0 | 1 | 1 | 1 |
| Feature4 | 0 | 1 | 1 | 1 |
| Feature5 | 1 | 1 | 0 | 1 |

**Weight (W 5,64)**

$\sigma$ (

| | son | Ronaldo | Messi | Dybala |
|---|---|---|---|---|
| son | 1/2 | 1/8 | 0 | 0 |
| Ronaldo | 1/8 | 1/16 | 1/12 | 1/12 |
| Messi | 0 | 1/12 | 1/9 | 1/9 |
| Dybala | 0 | 1/12 | 1/9 | 1/9 |

**Ahat(4,4)**

| | Filter1 | Filter2 | ... | Filter64 |
|---|---|---|---|---|
| son | 1 | 1 | 1 | 1 |
| Ronaldo | 1 | 1 | 0 | 1 |
| Messi | 0 | 1 | 0 | 1 |
| Dybala | 0 | 1 | 0 | 1 |

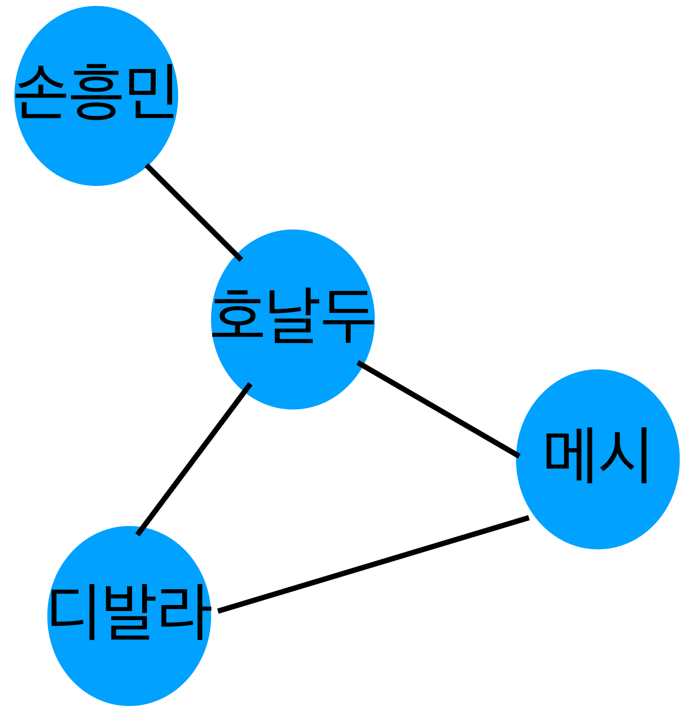
**XW (4,64)** 각 노드의 고차원 정보를 담은 행렬

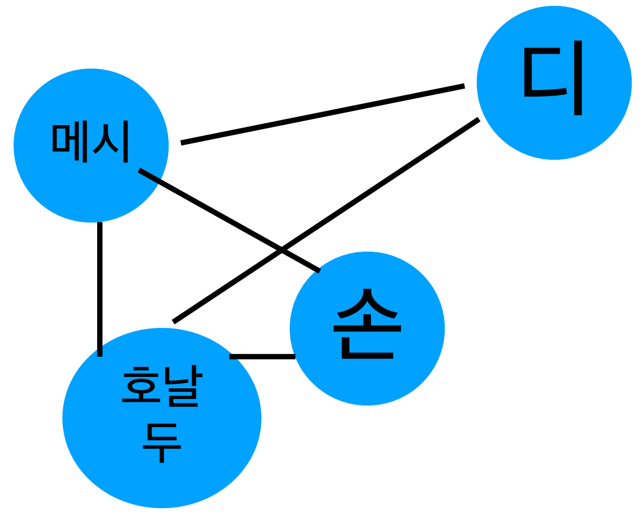
)

$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)} + b^{(l)}\right)$$

learnable parameters are shared

# 4. Graph Convolutional Networks

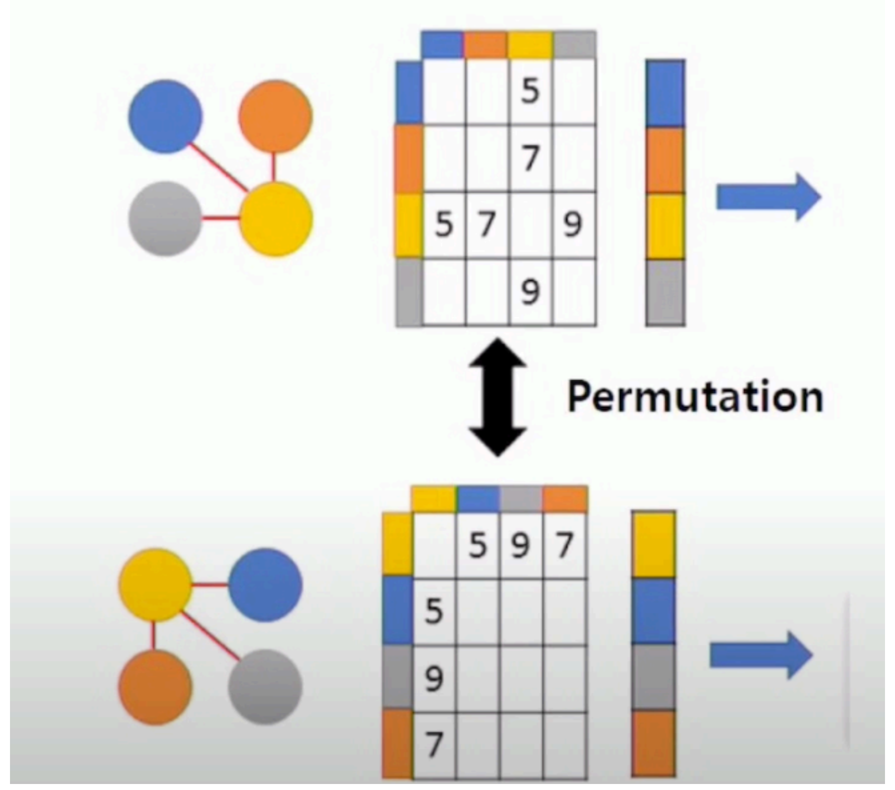



4.3 Readout Layer

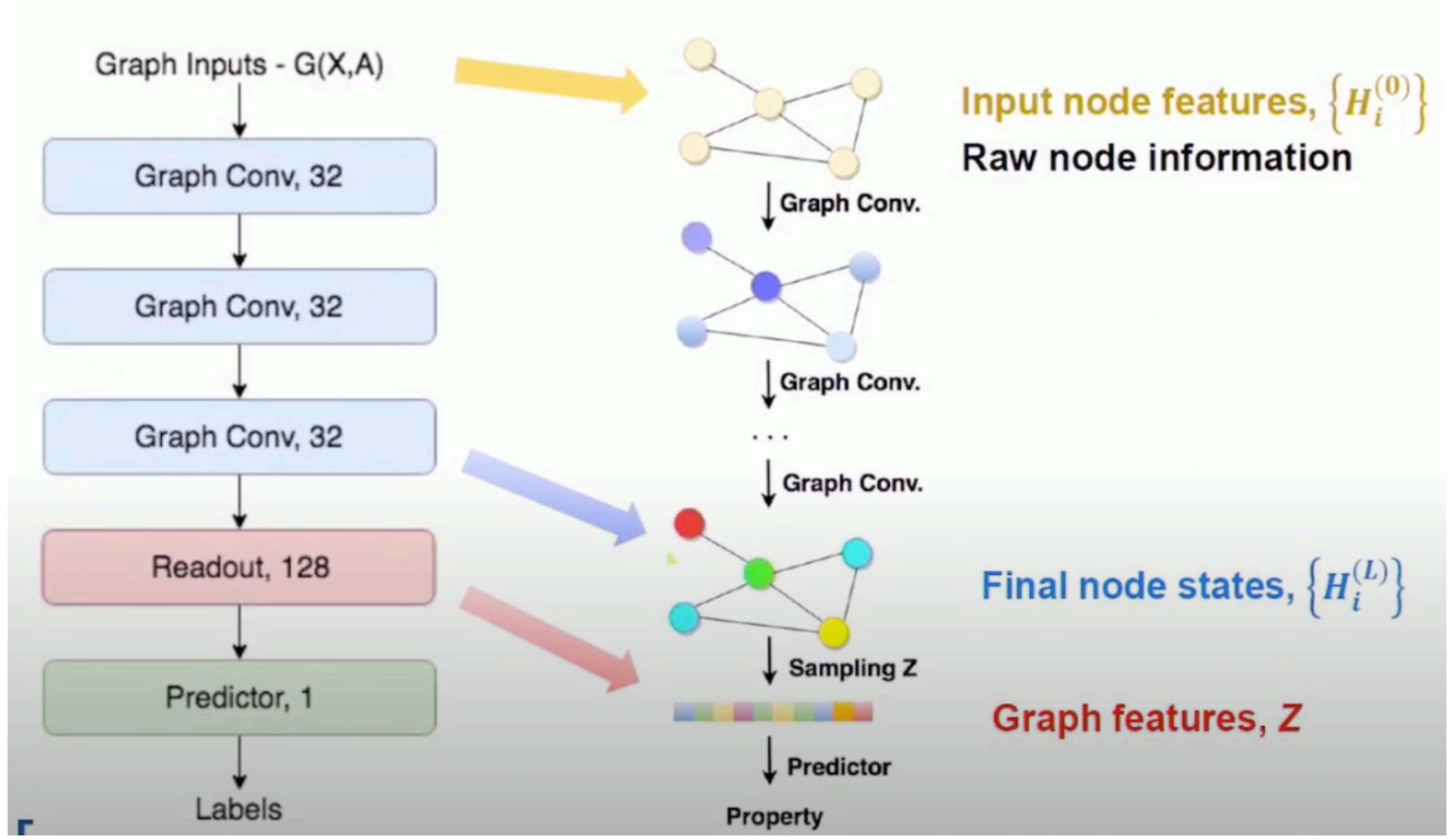
Permutation Invariance

Pytorch-geometric
stellargraph




**Node-wise summation**

$$z_G = \tau\left(\sum_{i \in G} MLP\left(H_i^{(L)}\right)\right)$$




- 노드의 순서가 바뀐다고 해서 그래프가 바뀌지는 않는다.
- 하지만 우리가 표현하는 **node feature matrix**는 그래프 모양이 바뀜에 따라 달라진다.
- 이것에 invariance를 부여하기 위해 readout layer를 거친다.

| | Feature1 | Feature2 | Feature3 |
|---|---|---|---|
| son | 0 | 1 | 0 |
| Ronaldo | 0 | 1 | 1 |
| Messi | 1 | 1 | 1 |
| Dybala | 1 | 1 | 1 |

| | Feature1 | Feature2 | Feature3 |
|---|---|---|---|
| Ronaldo | 0 | 1 | 1 |
| son | 0 | 1 | 1 |
| Messi | 1 | 1 | 1 |
| Dybala | 1 | 1 | 1 |

# References


[1] cs224w lecture1, 8
[2] [#30.Lec] Basic of Graph Convolution Network - 딥러닝 홀로서기
[3] A Comprehensive Survey on Graph Neural Networks 2019
[4] https://tykimos.github.io/warehouse/2019-7-4-ISS_2nd_Deep_Learning_Conference_All_Together_jtkim_file.pdf