# Graph Neural Networks for NLP

Zhaofeng Wu

# Outline

- Graphs in NLP and ML
- Graph neural networks (GNNs)
- Applying GNNs to linguistic graphs: Infusing Finetuning with Semantic Dependencies

# Graphs in NLP and ML

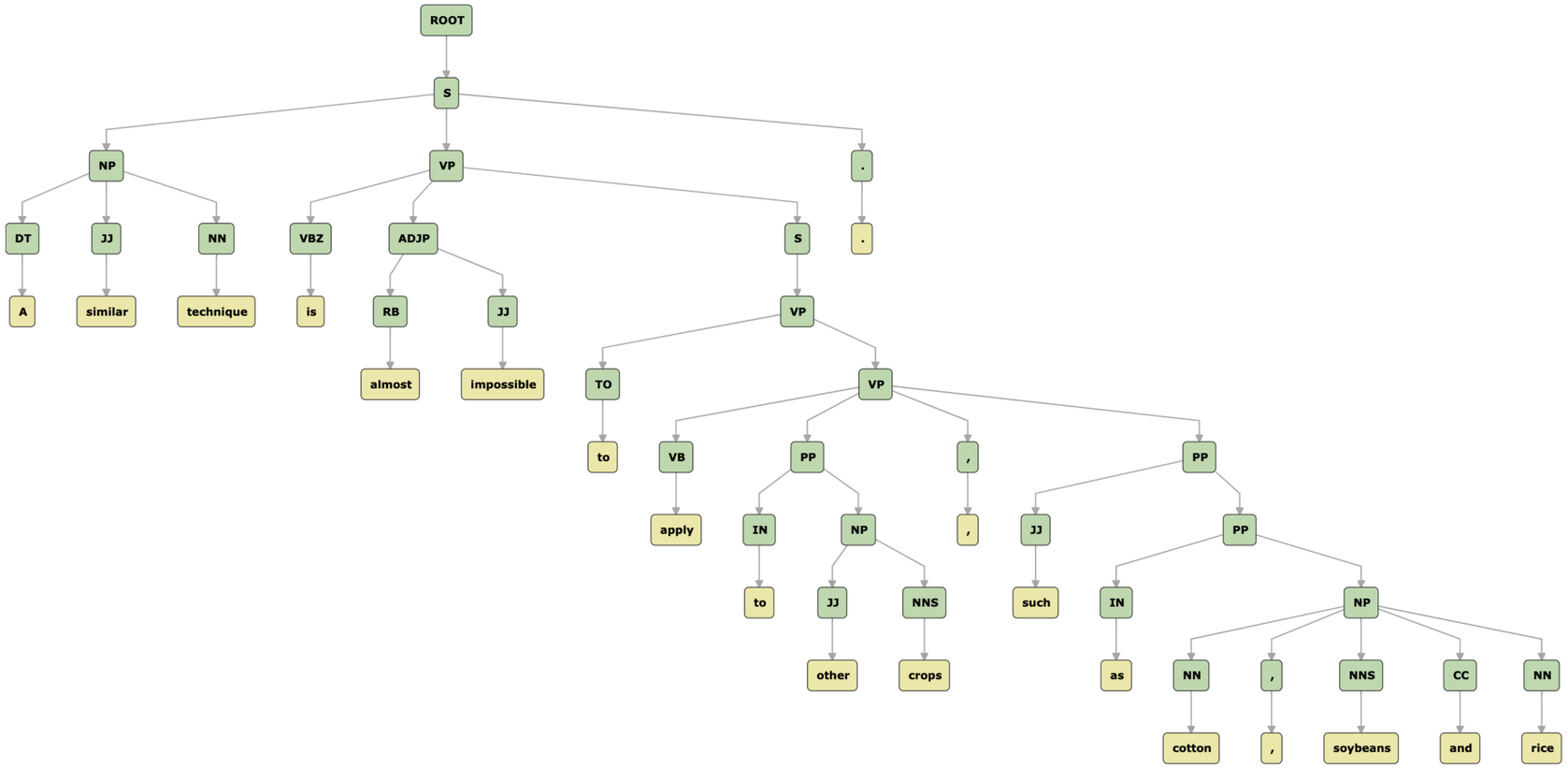
# Linguistic Structures

Running example: WSJ #20209013

“A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.”

# Linguistic Structures

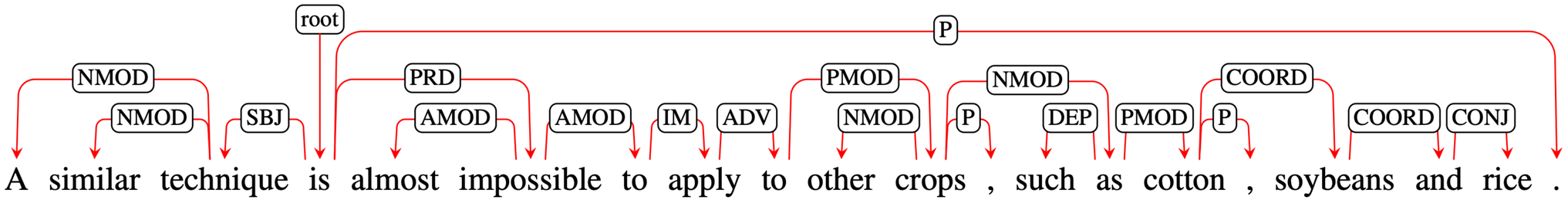
Running example: WSJ #20209013 “A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.”




Phrase structure tree; parsed by http://corenlp.run

# Linguistic Structures

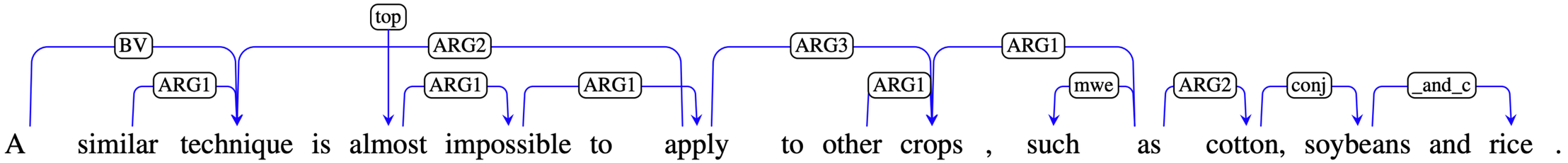
Running example: WSJ #20209013 “A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.”




Stanford Dependencies (UD) tree; from Ivanova (2012)

# Linguistic Structures

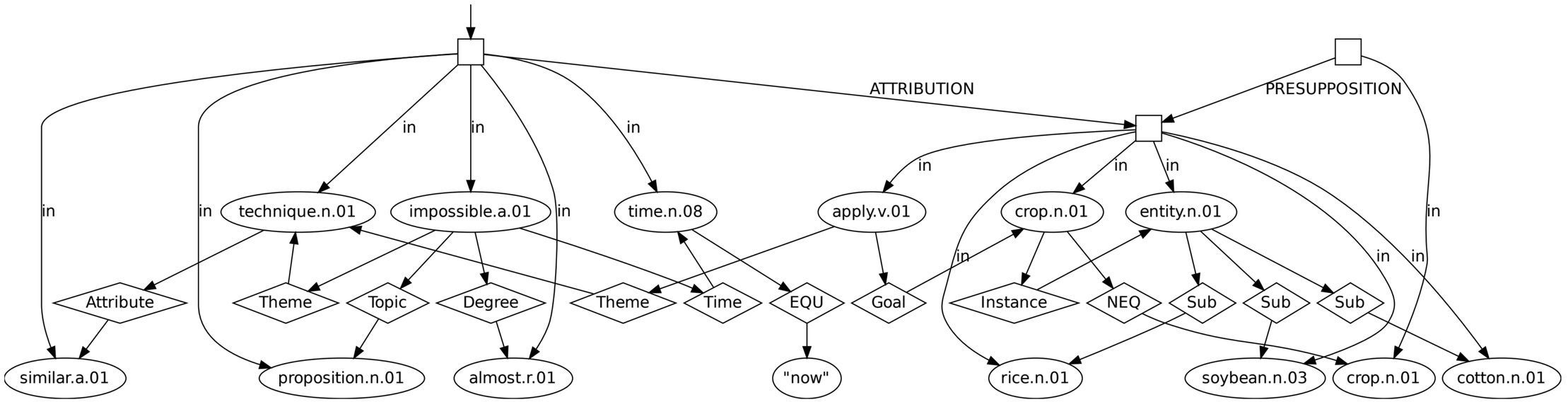
Running example: WSJ #20209013 “A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.”




DELPH-IN MRS-derived dependencies (DM) graph; from Oepen et al. (2015)

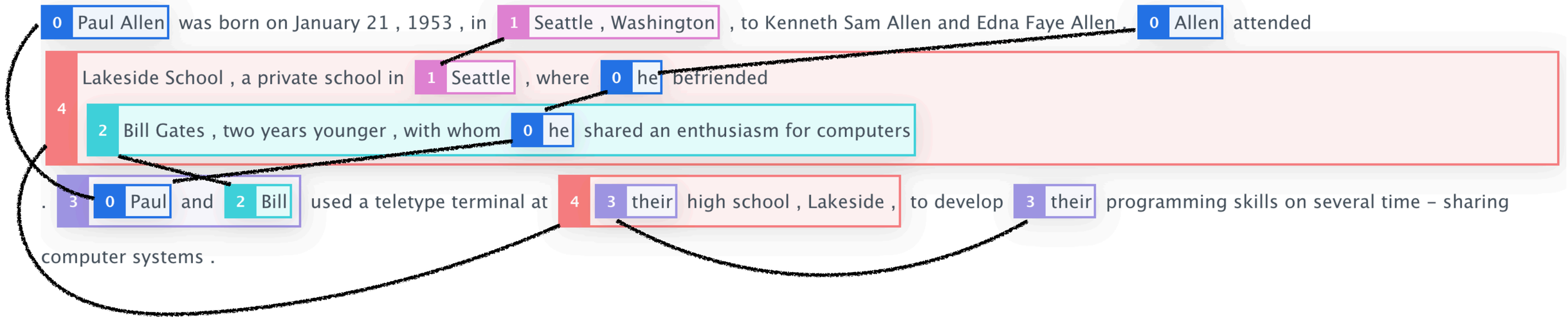
# Linguistic Structures

Running example: WSJ #20209013 “A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.”



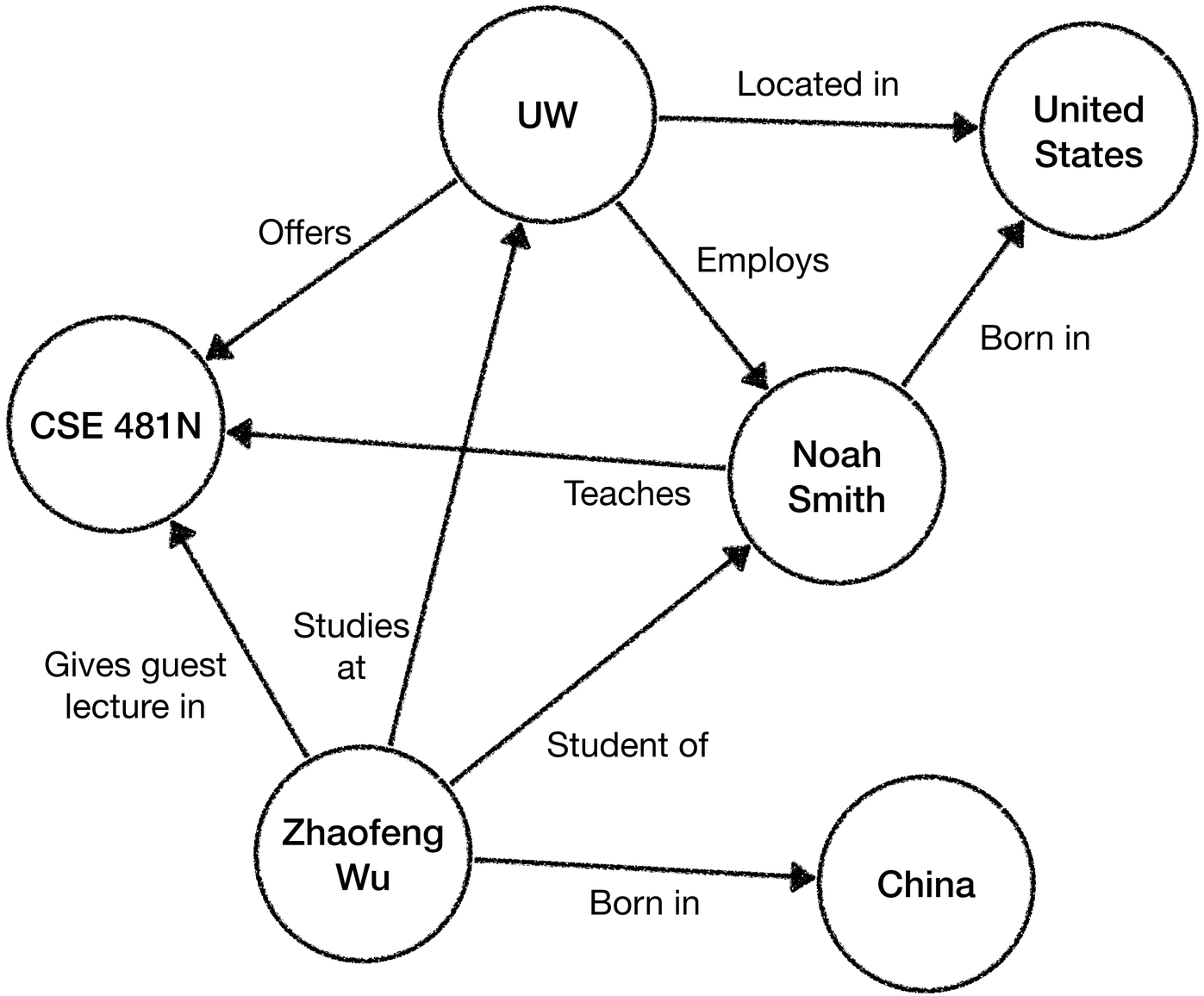

Discourse Representation Graph (DRG) graph; from Oepen et al. (2020)

# Linguistic Structures

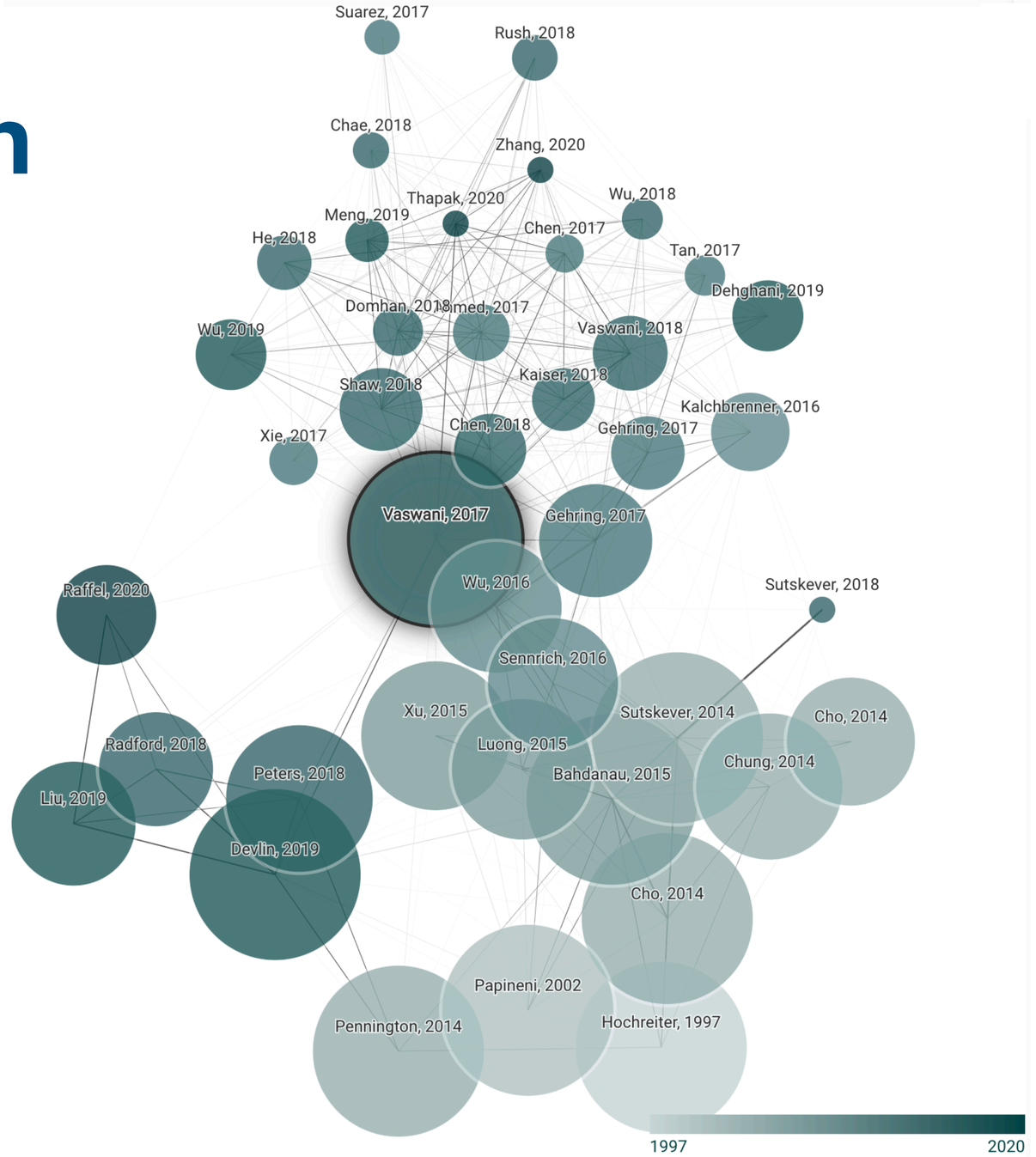



Coreference graph; from https://demo.allennlp.org/coreference-resolution

# Knowledge Graph

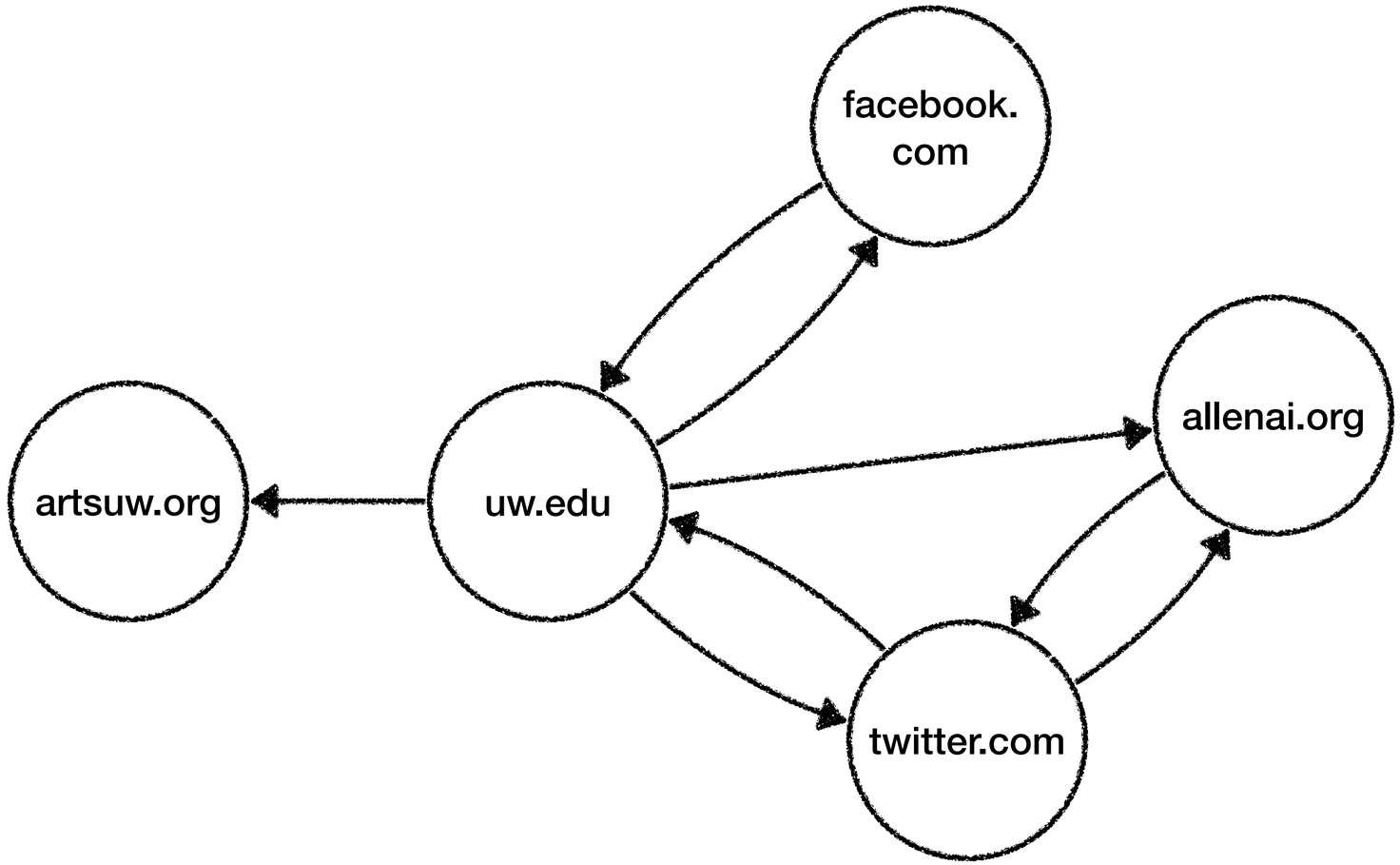


# Citation Graph




https://www.connectedpapers.com/main/204e3073870fae3d05bcbc2f6a8e263d9b72e776/Attention-is-All-you-Need/graph

# PageRank




“The founders of Google computed the Perron-Frobenius eigenvector of the web graph and became billionaires.”
— Preface to *Spectra of Graphs* by Brouwer and Haemers

# Graph Neural Networks

# Formulation

# Formulation

- Undirected $G = (V, E)$ with adjacency matrix $A \in \mathbb{R}^{n \times n}$; $n = |V|, m = |E|$

# Formulation

- Undirected $G = (V, E)$ with adjacency matrix $A \in \mathbb{R}^{n \times n}$; $n = |V|, m = |E|$
- Input
  - Node feature $X \in \mathbb{R}^{n \times d_v}$
  - (Sometimes) Edge feature $X^e \in \mathbb{R}^{m \times d_e}$

# Formulation

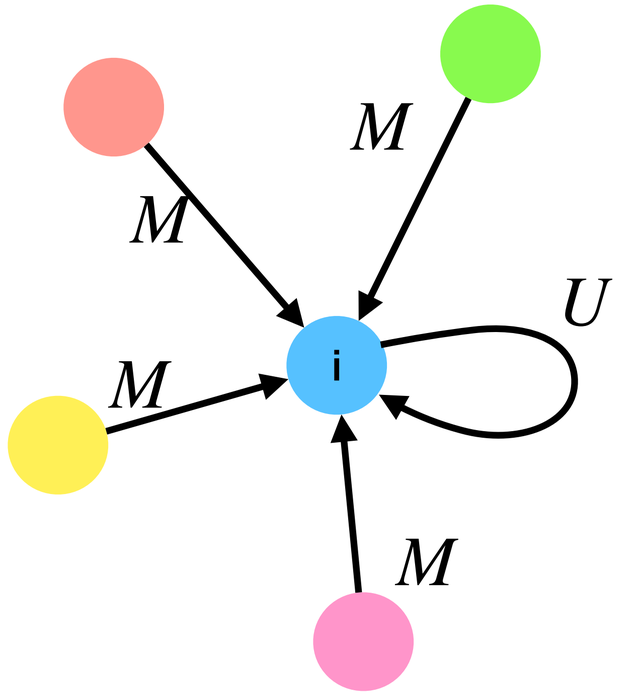
- Undirected $G = (V, E)$ with adjacency matrix $A \in \mathbb{R}^{n \times n}$; $n = |V|, m = |E|$
- Input
  - Node feature $X \in \mathbb{R}^{n \times d_v}$
  - (Sometimes) Edge feature $X^e \in \mathbb{R}^{m \times d_e}$
- Output: node representation $Z \in \mathbb{R}^{n \times d_o}$

# Formulation

- Undirected $G = (V, E)$ with adjacency matrix $A \in \mathbb{R}^{n \times n}$; $n = |V|, m = |E|$
- Input
  - Node feature $X \in \mathbb{R}^{n \times d_v}$
  - (Sometimes) Edge feature $X^e \in \mathbb{R}^{m \times d_e}$
- Output: node representation $Z \in \mathbb{R}^{n \times d_o}$
- How do we perform this update while leveraging the graph structure?

# Message Passing Framework; aka Spatial Methods

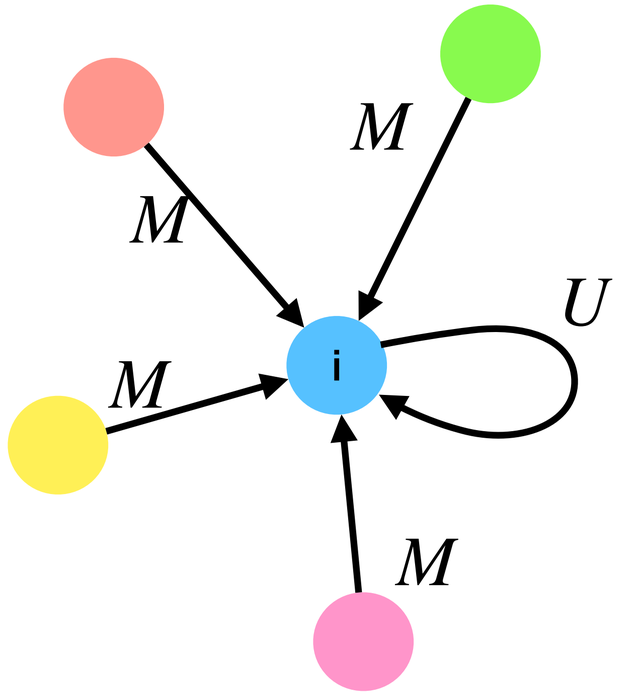
## Gilmer et al. (2018)



# Message Passing Framework; aka Spatial Methods

## Gilmer et al. (2018)

$$H^0 = X$$

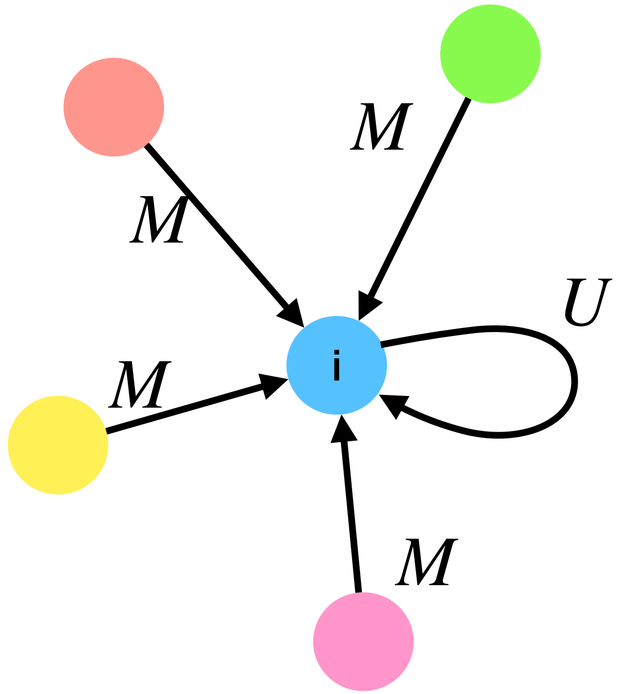


# Message Passing Framework; aka Spatial Methods

## Gilmer et al. (2018)

$$H^0 = X$$

$$\mathbf{m}_i^{l+1} = \sum_{j \in \mathcal{N}(i)} M_l(\mathbf{h}_i^l, \mathbf{h}_j^l, \mathbf{x}_{ij}^{e;l})$$



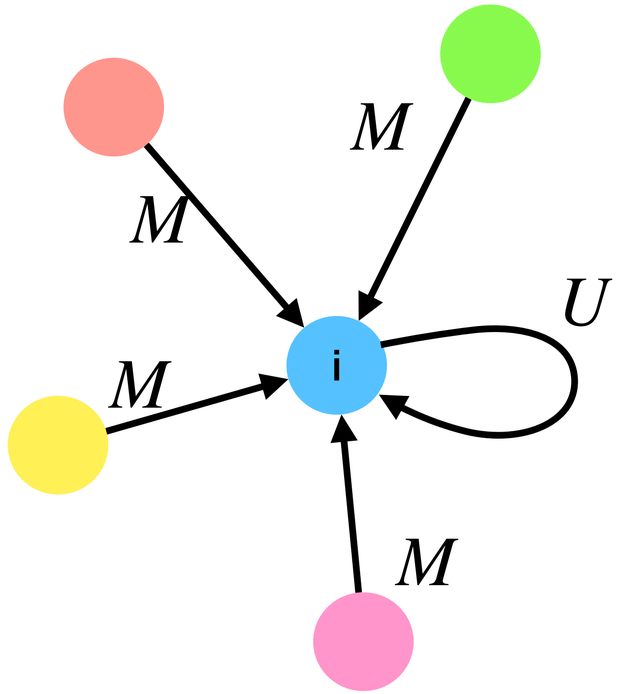
# Message Passing Framework; aka Spatial Methods

## Gilmer et al. (2018)

$$H^0 = X$$

$$\mathbf{m}_i^{l+1} = \sum_{j \in \mathcal{N}(i)} M_l(\mathbf{h}_i^l, \mathbf{h}_j^l, \mathbf{x}_{ij}^{e;l})$$

$$\mathbf{h}_i^{l+1} = U_t(\mathbf{h}_i^l, \mathbf{m}_i^{l+1})$$



# Message Passing Framework; aka Spatial Methods

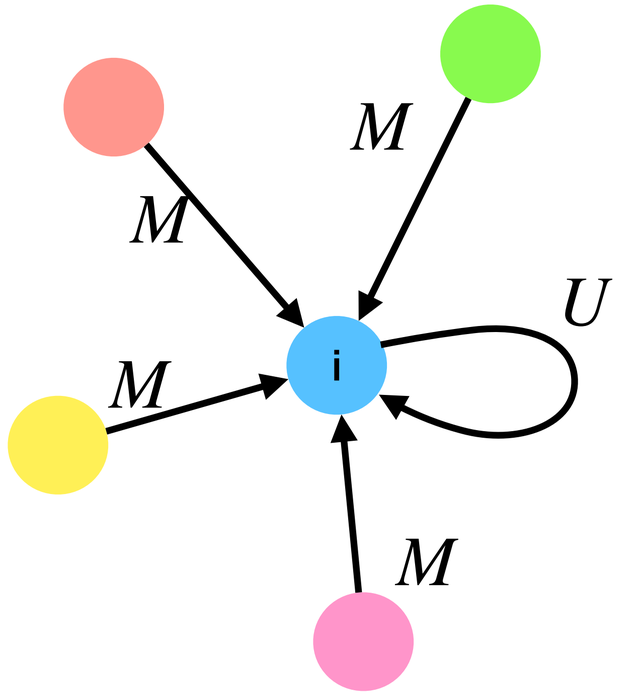
## Gilmer et al. (2018)

$$H^0 = X$$

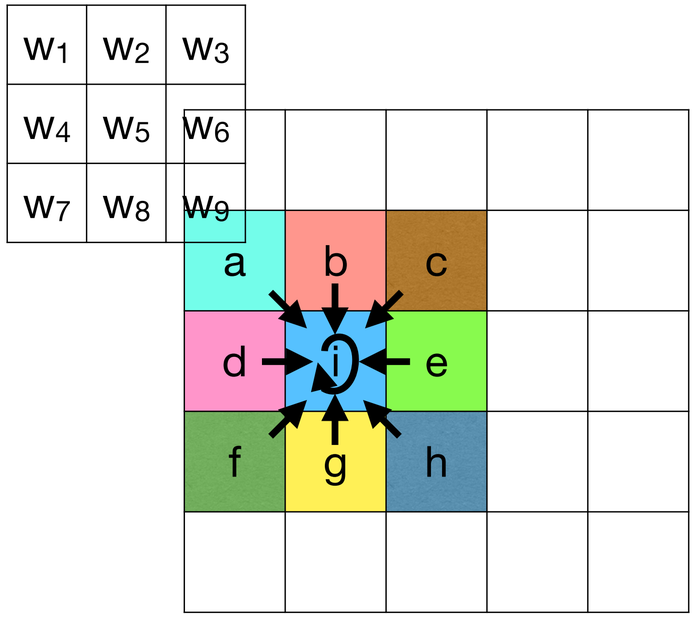
$$\mathbf{m}_i^{l+1} = \sum_{j \in \mathcal{N}(i)} M_l(\mathbf{h}_i^l, \mathbf{h}_j^l, \mathbf{x}_{ij}^{e;l})$$

$$\mathbf{h}_i^{l+1} = U_t(\mathbf{h}_i^l, \mathbf{m}_i^{l+1})$$

$$Z = H^L$$

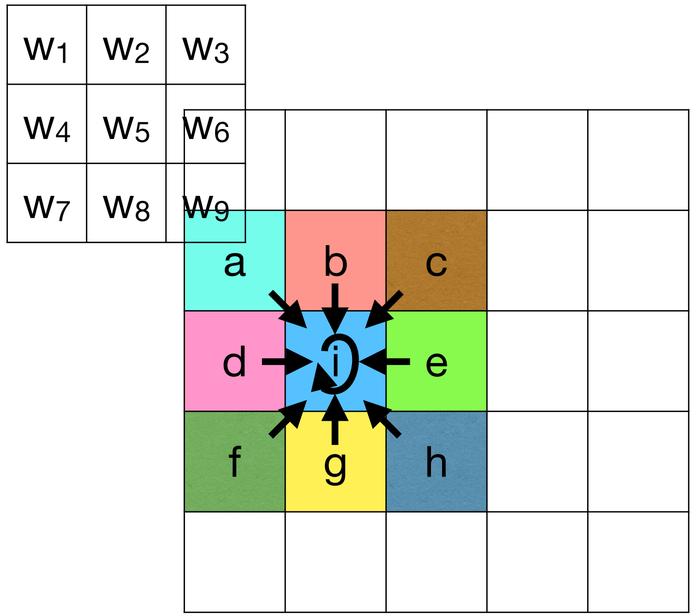


# Special Case: CNN as Message Passing



# Special Case: CNN as Message Passing

$$\mathbf{h}_i^{l+1} = w_1^l \mathbf{h}_a^l + w_2^l \mathbf{h}_b^l + \cdots + w_5^l \mathbf{h}_i^l + \cdots + w_9^l \mathbf{h}_h^l$$



# Graph Convolutional Networks (GCN)

## Kipf and Welling (2017)

# Graph Convolutional Networks (GCN)

## Kipf and Welling (2017)

Attempt #1: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i)} W^l \mathbf{h}_j^l\right)$

# Graph Convolutional Networks (GCN)

## Kipf and Welling (2017)

Attempt #1: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i)} W^l \mathbf{h}_j^l\right)$

Attempt #2: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} W^l \mathbf{h}_j^l\right)$

# Graph Convolutional Networks (GCN)

## Kipf and Welling (2017)

Attempt #1: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i)} W^l \mathbf{h}_j^l\right)$

Attempt #2: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} W^l \mathbf{h}_j^l\right)$

Attempt #3: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}} W^l \mathbf{h}_j^l\right)$

# Why “Convolutional”? The Short Answer

- Localized filters/kernels
- Translation invariance
- Hierarchical/multi-scale

# Why “Convolutional”? The Long Answer

# Why “Convolutional”? The Long Answer

- Message passing (spatial): $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}} W^l \mathbf{h}_j^l\right)$

# Why “Convolutional”? The Long Answer

- Message passing (spatial): $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}} W^l \mathbf{h}_j^l\right)$
- Matrix version: $H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l\right)$ where $\tilde{A} = A + I_n, \tilde{D} = \text{diag}\left(\sum_i \tilde{A}_{ii}\right)$

# Why “Convolutional”? The Long Answer

- Message passing (spatial): $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}} W^l \mathbf{h}_j^l\right)$
- Matrix version: $H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l\right)$ where $\tilde{A} = A + I_n, \tilde{D} = \text{diag}\left(\sum_i \tilde{A}_{ii}\right)$
- Motivation in Spectral Graph Theory

# Why “Convolutional”? The Long Answer

- Message passing (spatial): $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}} W^l \mathbf{h}_j^l\right)$
- Matrix version: $H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l\right)$ where $\tilde{A} = A + I_n, \tilde{D} = \text{diag}\left(\sum_i \tilde{A}_{ii}\right)$
- Motivation in Spectral Graph Theory
  - Spectral decomposition of the normalized graph Laplacian $L = I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = U\Lambda U^\top$

# Why “Convolutional”? The Long Answer

- Message passing (spatial): $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}} W^l \mathbf{h}_j^l\right)$
- Matrix version: $H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l\right)$ where $\tilde{A} = A + I_n, \tilde{D} = \text{diag}\left(\sum_i \tilde{A}_{ii}\right)$
- Motivation in Spectral Graph Theory
  - Spectral decomposition of the normalized graph Laplacian
    $L = I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = U\Lambda U^\top$
  - Generalizing convolution to “graph convolution”:
    $\mathbf{h} *_G \mathbf{g} = F^{-1}(F(\mathbf{h}) \odot F(\mathbf{g})), F(\mathbf{h}) = U\mathbf{h}$

# Why “Convolutional”? The Long Answer

- Message passing (spatial): $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}} W^l \mathbf{h}_j^l\right)$
- Matrix version: $H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l\right)$ where $\tilde{A} = A + I_n, \tilde{D} = \text{diag}\left(\sum_i \tilde{A}_{ii}\right)$
- Motivation in Spectral Graph Theory
  - Spectral decomposition of the normalized graph Laplacian
    $L = I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = U\Lambda U^\top$
  - Generalizing convolution to “graph convolution”:
    $\mathbf{h} *_G \mathbf{g} = F^{-1}(F(\mathbf{h}) \odot F(\mathbf{g})), F(\mathbf{h}) = U\mathbf{h}$
  - Approximation leads to the matrix version above

# I have node embeddings. Now what?

# I have node embeddings. Now what?

- Node classification: $\text{softmax}(\mathbf{z}_i)$

# I have node embeddings. Now what?

- Node classification: $\text{softmax}(\mathbf{z}_i)$
- Graph classification: $\text{softmax}\left(\frac{1}{n}\sum_i \mathbf{z}_i\right)$

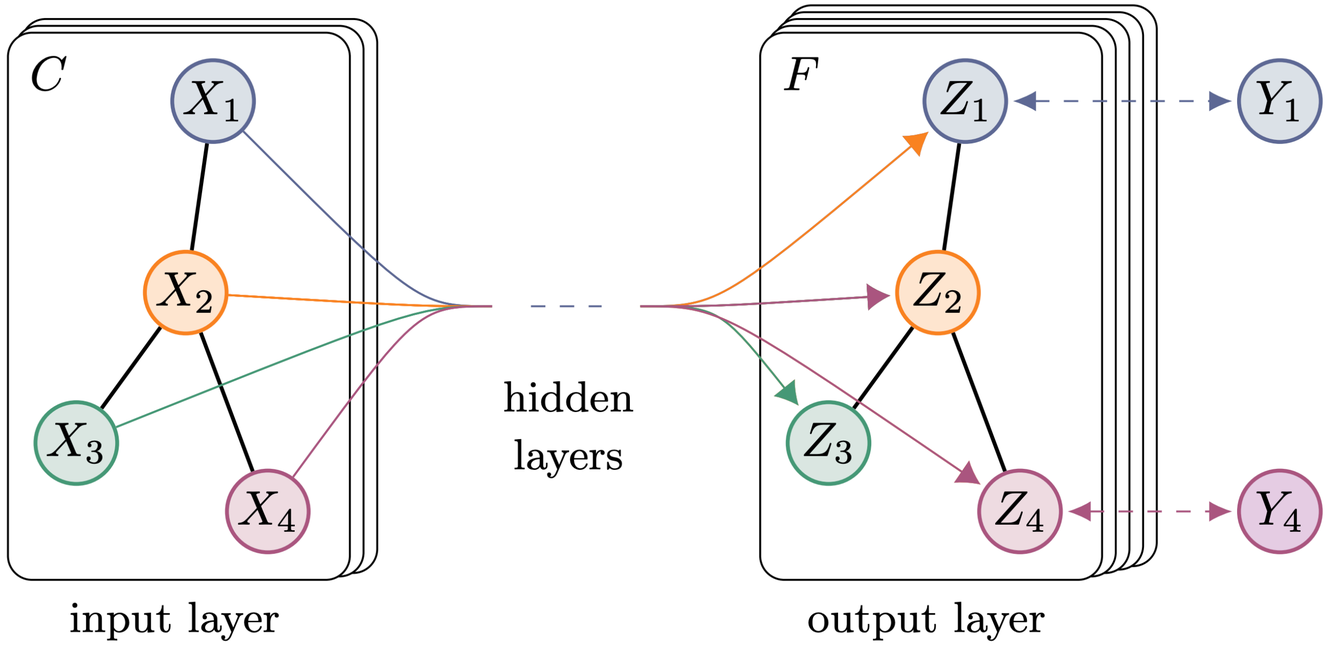
# I have node embeddings. Now what?

- Node classification: $\text{softmax}(\mathbf{z}_i)$
- Graph classification: $\text{softmax}\left(\frac{1}{n}\sum_i \mathbf{z}_i\right)$
- Link prediction: $p(A_{ij}) = \text{sigmoid}(\mathbf{z}_i^\top \mathbf{z}_j)$

# GCN for Semi-Supervised Learning

## Kipf and Welling (2017)

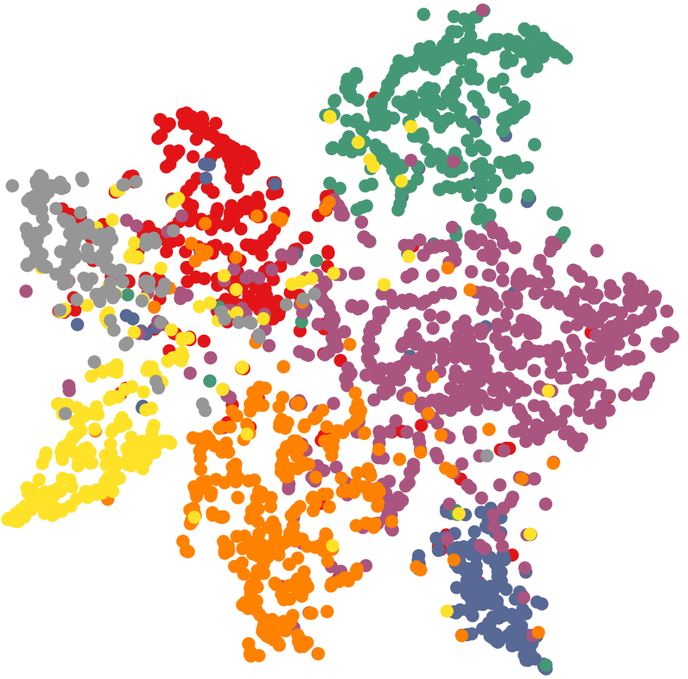
- Citation networks with partially labeled nodes



# GCN for Semi-Supervised Learning

## Kipf and Welling (2017)

| Method | Citeseer | Cora | Pubmed | NELL |
|---|---|---|---|---|
| ManiReg [3] | 60.1 | 59.5 | 70.7 | 21.8 |
| SemiEmb [28] | 59.6 | 59.0 | 71.1 | 26.7 |
| LP [32] | 45.3 | 68.0 | 63.0 | 26.5 |
| DeepWalk [22] | 43.2 | 67.2 | 65.3 | 58.1 |
| ICA [18] | 69.1 | 75.1 | 73.9 | 23.1 |
| Planetoid* [29] | 64.7 (26s) | 75.7 (13s) | 77.2 (25s) | 61.9 (185s) |
| **GCN** (this paper) | **70.3** (7s) | **81.5** (4s) | **79.0** (38s) | **66.0** (48s) |
| GCN (rand. splits) | $67.9 \pm 0.5$ | $80.1 \pm 0.5$ | $78.9 \pm 0.7$ | $58.4 \pm 1.7$ |



t-SNE of hidden layer activation.

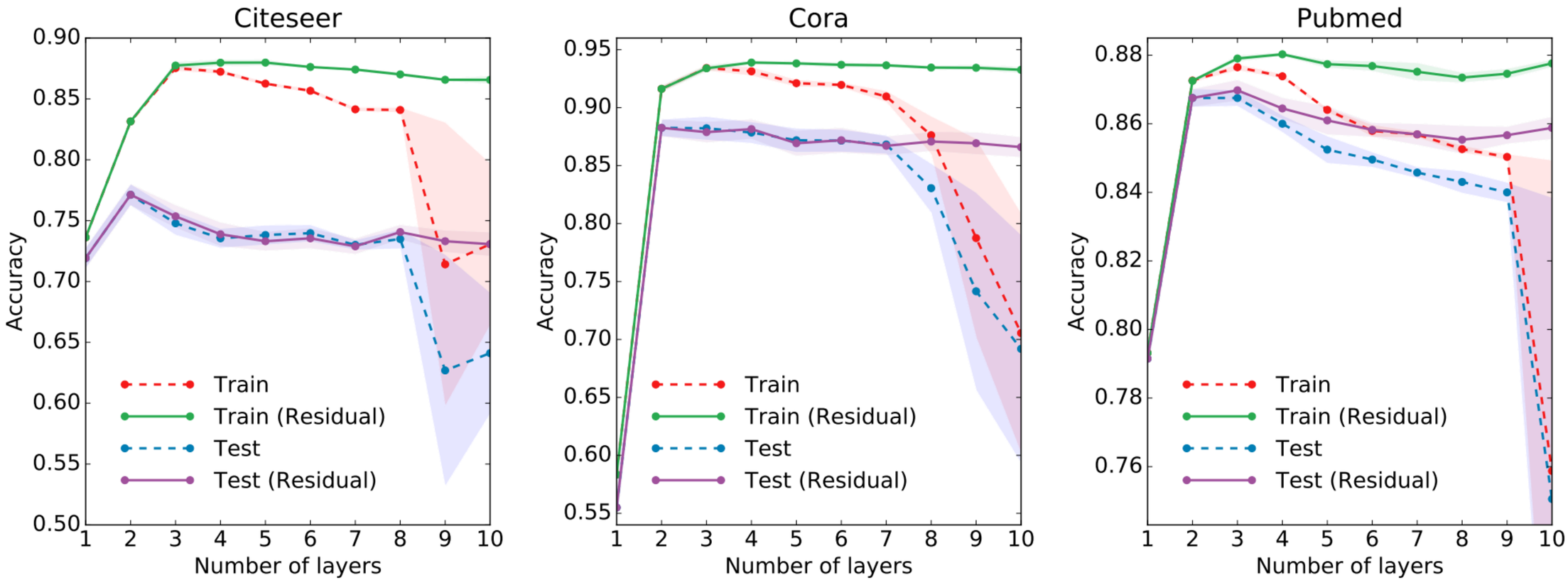
# GCN for Semi-Supervised Learning

## Kipf and Welling (2017)

| | Description | | Propagation model | Citeseer | Cora | Pubmed |
|---|---|---|---|---|---|---|
| | Chebyshev filter (Eq. 5) | $K = 3$ | $\sum_{k=0}^{K} T_k(\tilde{L}) X \Theta_k$ | 69.8 | 79.5 | 74.4 |
| | | $K = 2$ | | 69.6 | 81.2 | 73.8 |
| | $1^{st}$-order model (Eq. 6) | | $X\Theta_0 + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X \Theta_1$ | 68.3 | 80.0 | 77.5 |
| | Single parameter (Eq. 7) | | $(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) X \Theta$ | 69.3 | 79.2 | 77.4 |
| ➡ | **Renormalization trick** (Eq. 8) | | $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta$ | **70.3** | **81.5** | **79.0** |
| | $1^{st}$-order term only | | $D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X \Theta$ | 68.7 | 80.5 | 77.8 |
| ➡ | Multi-layer perceptron | | $X\Theta$ | 46.5 | 55.1 | 71.4 |

# GCN for Semi-Supervised Learning

## Kipf and Welling (2017)



# Relational Graph Convolutional Networks (RGCN)

## Schlichtkrull et al. (2018)

# Relational Graph Convolutional Networks (RGCN)

## Schlichtkrull et al. (2018)

- Relations (edge types) $\mathscr{R}$

# Relational Graph Convolutional Networks (RGCN)

## Schlichtkrull et al. (2018)

- Relations (edge types) $\mathscr{R}$
- GCN: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathscr{N}(i) \cup \{i\}} \frac{1}{c_{ij}} W^l \mathbf{h}_j^l\right)$ where $c_{ij} = \sqrt{|\mathscr{N}(i)| + 1}\sqrt{|\mathscr{N}(j)| + 1}$

# Relational Graph Convolutional Networks (RGCN)

## Schlichtkrull et al. (2018)

- Relations (edge types) $\mathscr{R}$
- GCN: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{c_{ij}} W^l \mathbf{h}_j^l\right)$ where $c_{ij} = \sqrt{|\mathcal{N}(i)| + 1}\sqrt{|\mathcal{N}(j)| + 1}$
- RGCN: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{r \in \mathscr{R}} \sum_{j \in \mathcal{N}^r(i)} \frac{1}{c_{ijr}} W_r^l \mathbf{h}_j^l\right)$ where $c_{ijr} = |\mathcal{N}^r(i)|$
  - Assume $\mathscr{R}$ and $\mathcal{N}^r$ capture self-loop

# Relational Graph Convolutional Networks (RGCN)

## Schlichtkrull et al. (2018)

- Relations (edge types) $\mathscr{R}$
- GCN: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{c_{ij}} W^l \mathbf{h}_j^l\right)$ where $c_{ij} = \sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}$
- RGCN: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{r \in \mathscr{R}} \sum_{j \in \mathcal{N}^r(i)} \frac{1}{c_{ijr}} W_r^l \mathbf{h}_j^l\right)$ where $c_{ijr} = |\mathcal{N}^r(i)|$
  - Assume $\mathscr{R}$ and $\mathcal{N}^r$ capture self-loop
- Can be used to model directed graphs

# RGCN: Regularization

## Schlichtkrull et al. (2018)

# RGCN: Regularization

## Schlichtkrull et al. (2018)

- Basis-decomposition: $W_r^l = \sum_{b=1}^{B} a_{rb}^l V_b^l$

# RGCN: Regularization

## Schlichtkrull et al. (2018)

- Basis-decomposition: $W_r^l = \sum_{b=1}^{B} a_{rb}^l V_b^l$
- Block-diagonal-decomposition: $W_r^l = \bigoplus_{b=1}^{B} Q_{br}^l$

# Graph Attention Networks (GAT)

## Velickovič et al. (2018)

# Graph Attention Networks (GAT)

## Velickovič et al. (2018)

- GCN: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{c_{ij}} W^l \mathbf{h}_j^l\right)$ where $c_{ij} = \sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}$

# Graph Attention Networks (GAT)

## Velickovič et al. (2018)

- GCN: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{c_{ij}} W^l \mathbf{h}_j^l\right)$ where $c_{ij} = \sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}$
- GAT: $\mathbf{h}_i^{l+1} = \|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{k;l} W^{k;l} \mathbf{h}_j^l\right)$ where $\alpha_{ij} = \mathrm{softmax}_j(a(\mathbf{h}_i, \mathbf{h}_j))$

# Graph Attention Networks (GAT)

## Velickovič et al. (2018)

- GCN: $\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{c_{ij}} W^l \mathbf{h}_j^l\right)$ where $c_{ij} = \sqrt{|\mathcal{N}(i)|+1}\sqrt{|\mathcal{N}(j)|+1}$
- GAT: $\mathbf{h}_i^{l+1} = \|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{k;l} W^{k;l} \mathbf{h}_j^l\right)$ where $\alpha_{ij} = \mathrm{softmax}_j(a(\mathbf{h}_i, \mathbf{h}_j))$
  - They used $a(\mathbf{h}_i, \mathbf{h}_j) = \mathrm{LeakyReLU}(\mathbf{a}^\top [W\mathbf{h}_i \| W\mathbf{h}_j])$

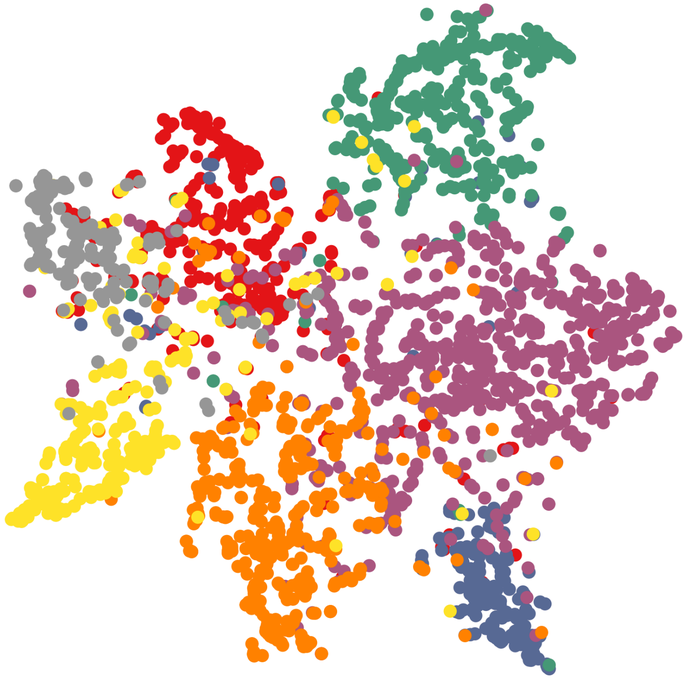
# GAT for Semi-Supervised Learning

## Velickovič et al. (2018)

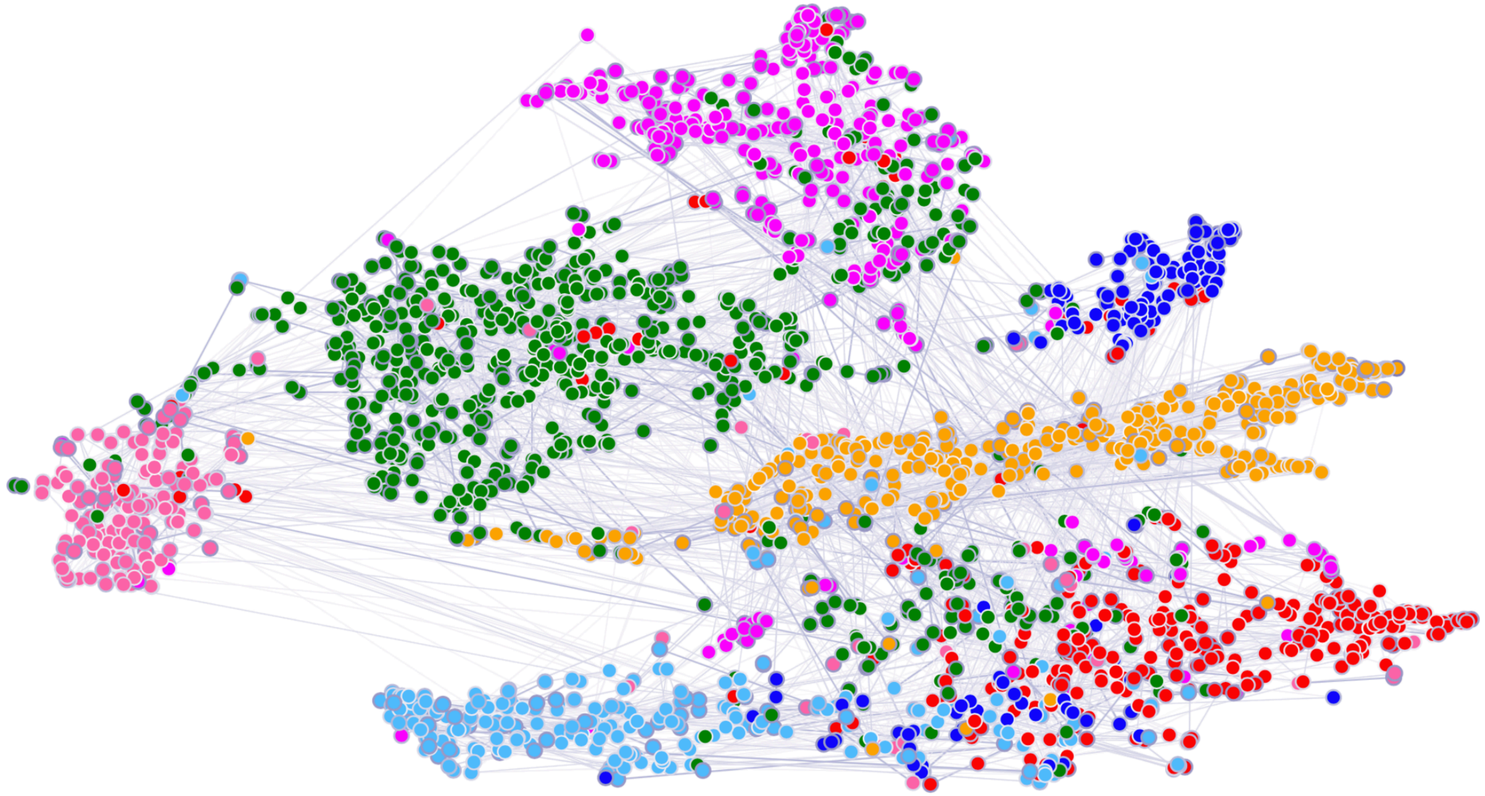
| Method | Cora | Citeseer | Pubmed |
|---|---|---|---|
| MLP | 55.1% | 46.5% | 71.4% |
| ManiReg (Belkin et al., 2006) | 59.5% | 60.1% | 70.7% |
| SemiEmb (Weston et al., 2012) | 59.0% | 59.6% | 71.7% |
| LP (Zhu et al., 2003) | 68.0% | 45.3% | 63.0% |
| DeepWalk (Perozzi et al., 2014) | 67.2% | 43.2% | 65.3% |
| ICA (Lu & Getoor, 2003) | 75.1% | 69.1% | 73.9% |
| Planetoid (Yang et al., 2016) | 75.7% | 64.7% | 77.2% |
| Chebyshev (Defferrard et al., 2016) | 81.2% | 69.8% | 74.4% |
| GCN (Kipf & Welling, 2017) | 81.5% | 70.3% | **79.0%** |
| MoNet (Monti et al., 2016) | 81.7 ± 0.5% | — | 78.8 ± 0.3% |
| GCN-64* | 81.4 ± 0.5% | 70.9 ± 0.5% | **79.0** ± 0.3% |
| **GAT** (ours) | **83.0** ± 0.7% | **72.5** ± 0.7% | **79.0** ± 0.3% |

# GAT for Semi-Supervised Learning

## Velickovič et al. (2018)

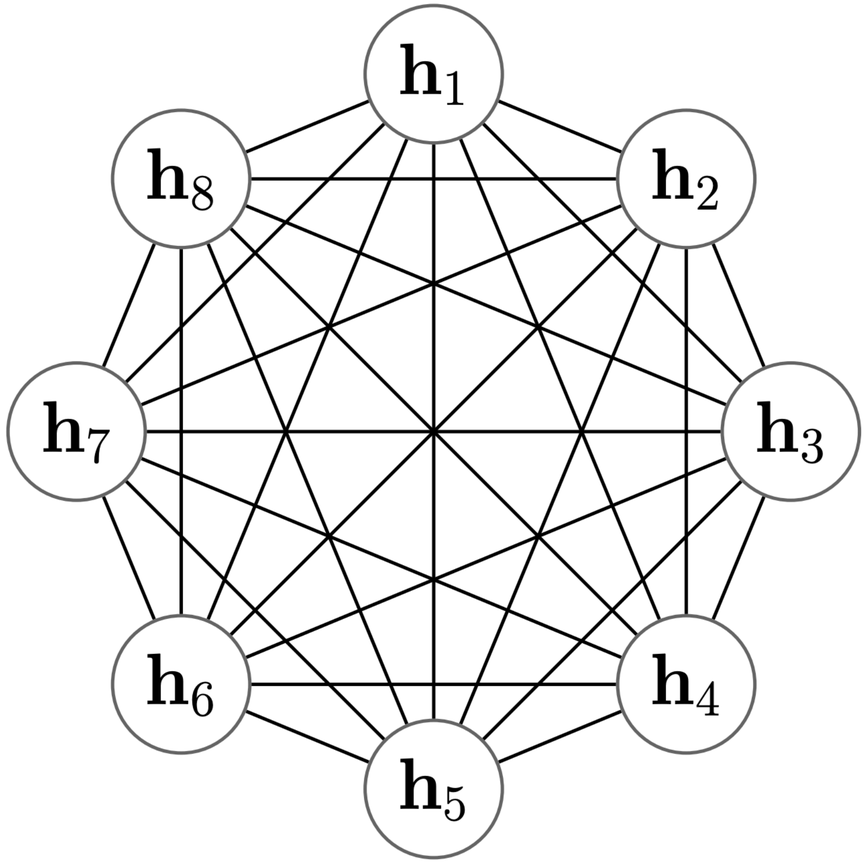


GCN



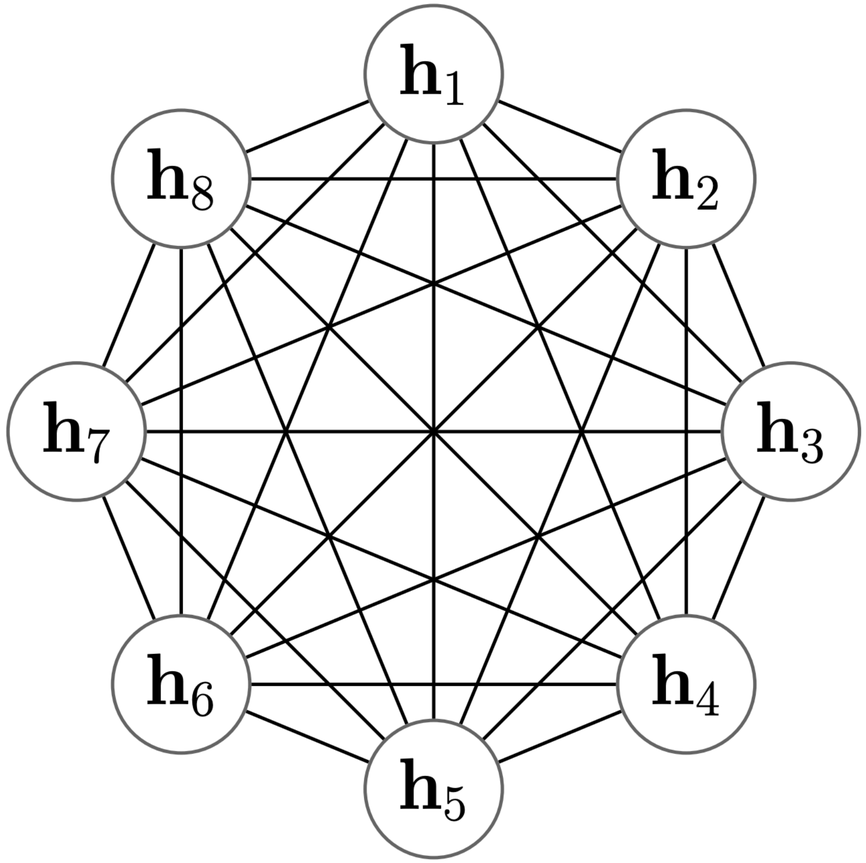
GAT

# Special Case: Transformer as Message Passing

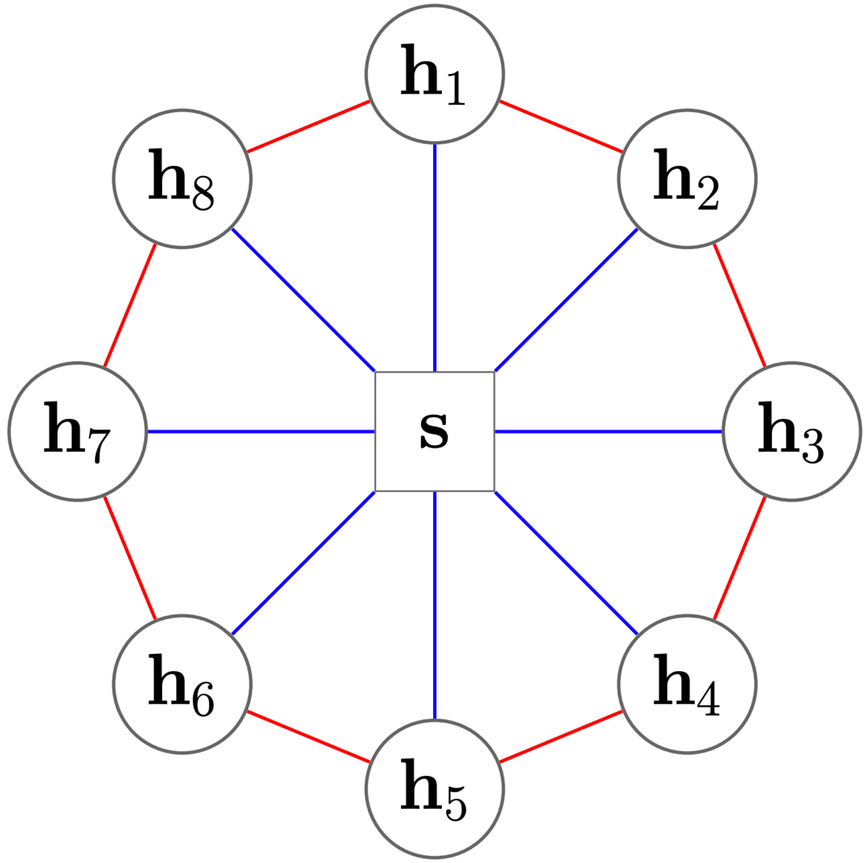



Transformer. From Guo et al. (2018).

# Special Case: Transformer as Message Passing




Transformer. From Guo et al. (2018).



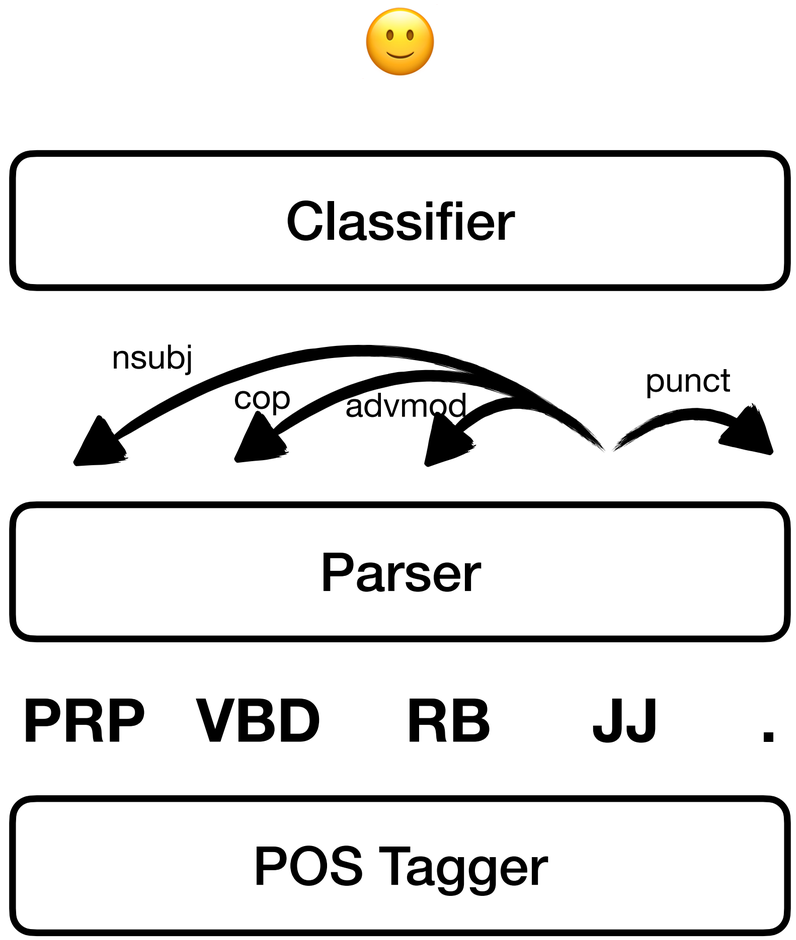

Star-Transformer. From Guo et al. (2018).

# Applying GNNs to Linguistic Graphs

# Infusing Finetuning with Semantic Dependencies

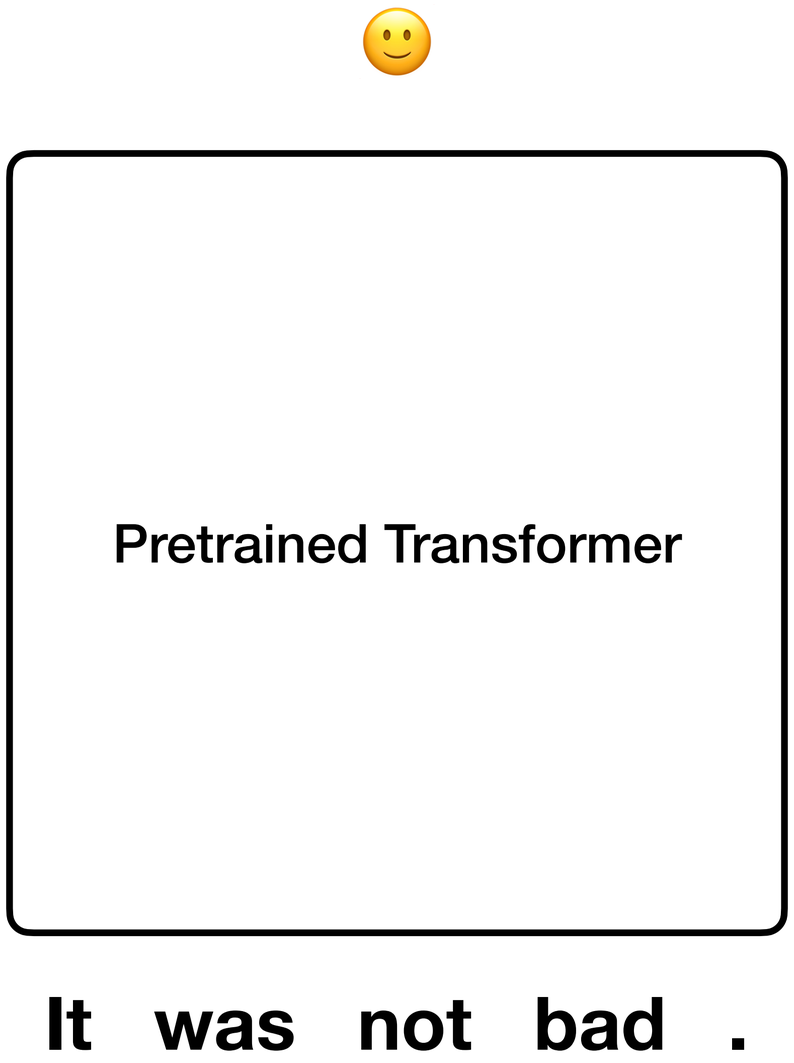
Zhaofeng Wu, Hao Peng, and Noah Smith. TACL 2021.

# Motivation


Classifier
nsubj
cop
advmod
punct
Parser
PRP VBD RB JJ .
POS Tagger
It was not bad .

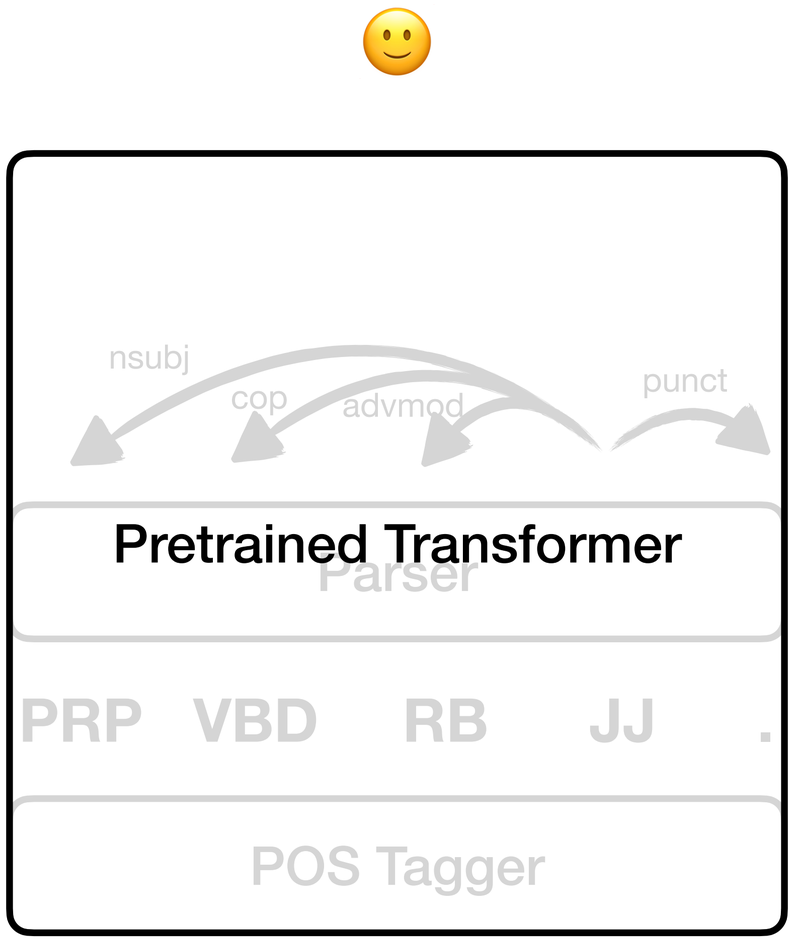
Devlin et al. (2018); Liu et al. (2019b)

# Motivation



Hewitt and Manning (2019); Tenney et al. (2019); Liu et al. (2019a)

# Motivation



# Would semantics help?

# Introduction

# Introduction

- We show BERT/RoBERTa less prominently surface semantics…

# Introduction

- We show BERT/RoBERTa less prominently surface semantics…
- … and the explicit incorporation of semantic information:

# Introduction

- We show BERT/RoBERTa less prominently surface semantics…
- … and the explicit incorporation of semantic information:

  1. Improves downstream task performance

# Introduction

- We show BERT/RoBERTa less prominently surface semantics…
- … and the explicit incorporation of semantic information:

  1. Improves downstream task performance

  2. Helps guard against frequent yet invalid heuristics

# Introduction

- We show BERT/RoBERTa less prominently surface semantics…
- … and the explicit incorporation of semantic information:

  1. Improves downstream task performance
  2. Helps guard against frequent yet invalid heuristics
  3. Better captures nuanced linguistic phenomena

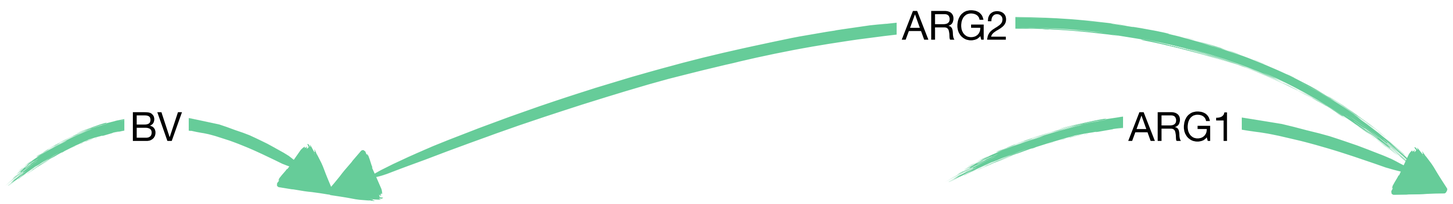
# Introduction

- We show BERT/RoBERTa less prominently surface semantics…
- … and the explicit incorporation of semantic information:

  1. Improves downstream task performance
  2. Helps guard against frequent yet invalid heuristics
  3. Better captures nuanced linguistic phenomena
  4. Increases training sample efficiency

Adapted from WSJ #20209013

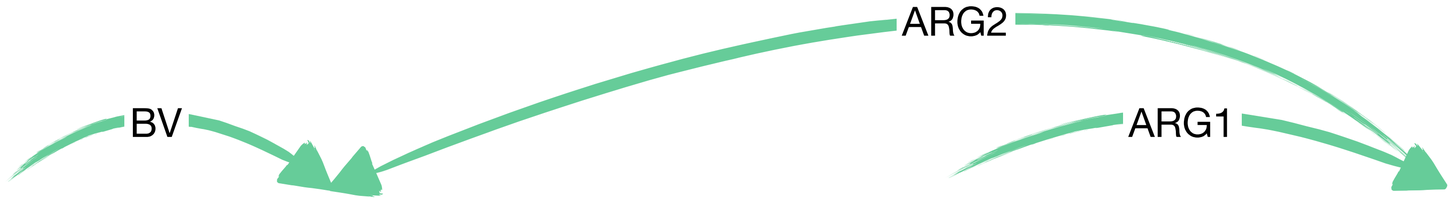
# Operationalizing “Meaning”



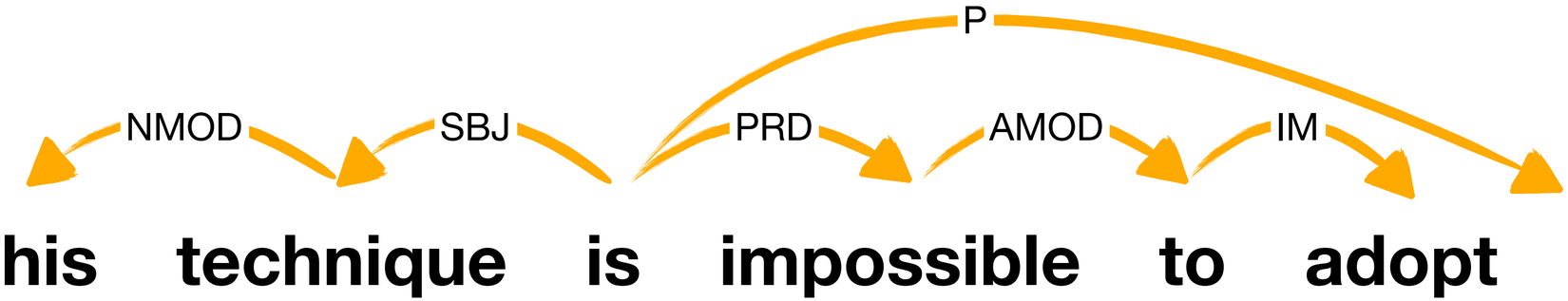

**This technique is impossible to adopt .**

DELPH-IN MRS-Derived Dependencies (**DM**; Ivanova et al., 2012)

# Operationalizing “Meaning”

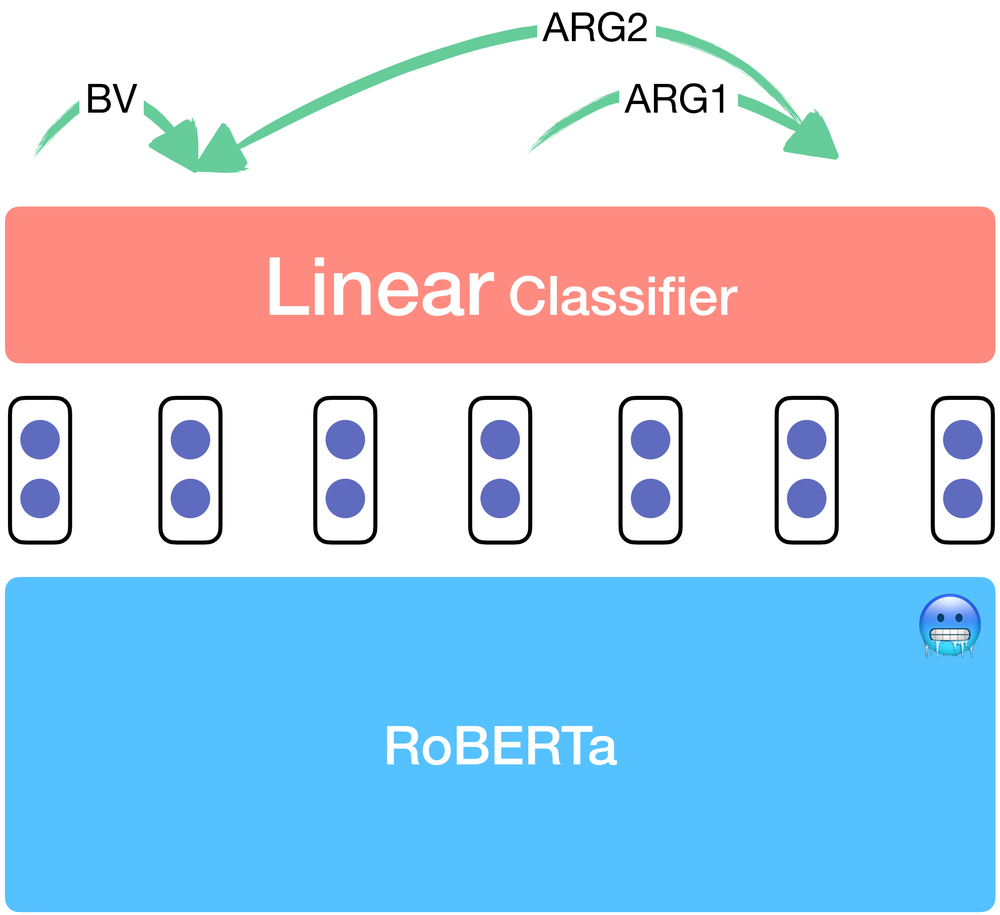



DELPH-IN MRS-Derived Dependencies (**DM**; Ivanova et al., 2012)




Stanford Dependencies (**SD**; de Marneffe et al., 2006)

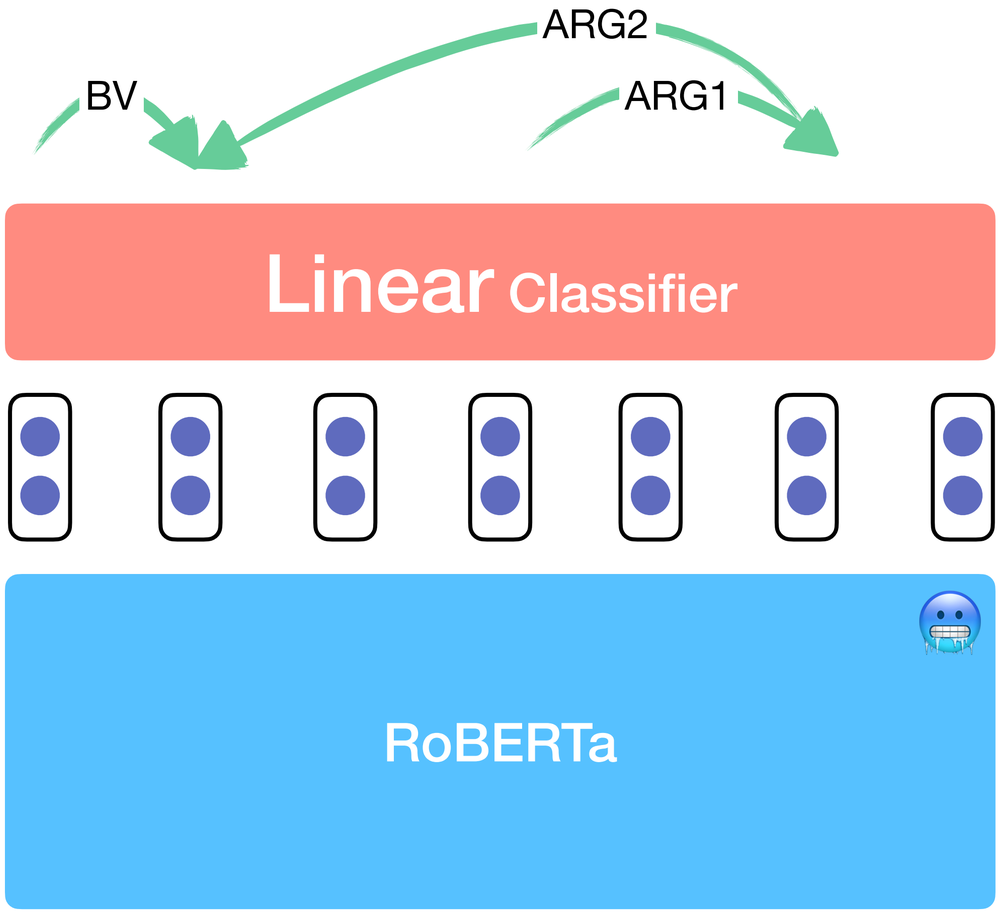
# Probing RoBERTa with Semantics




**This technique is impossible to adopt .**

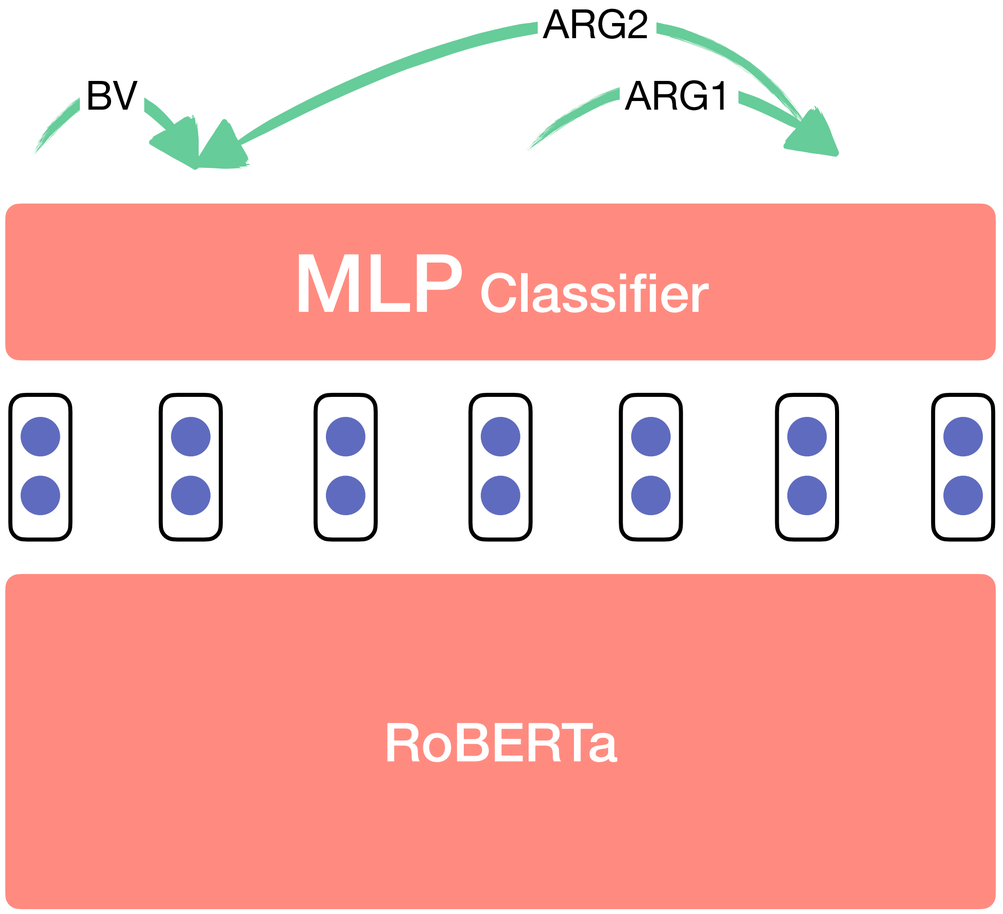
**Probing** model (Shi et al., 2016; Adi et al., 2017)

# Probing RoBERTa with Semantics




**This technique is impossible to adopt .**

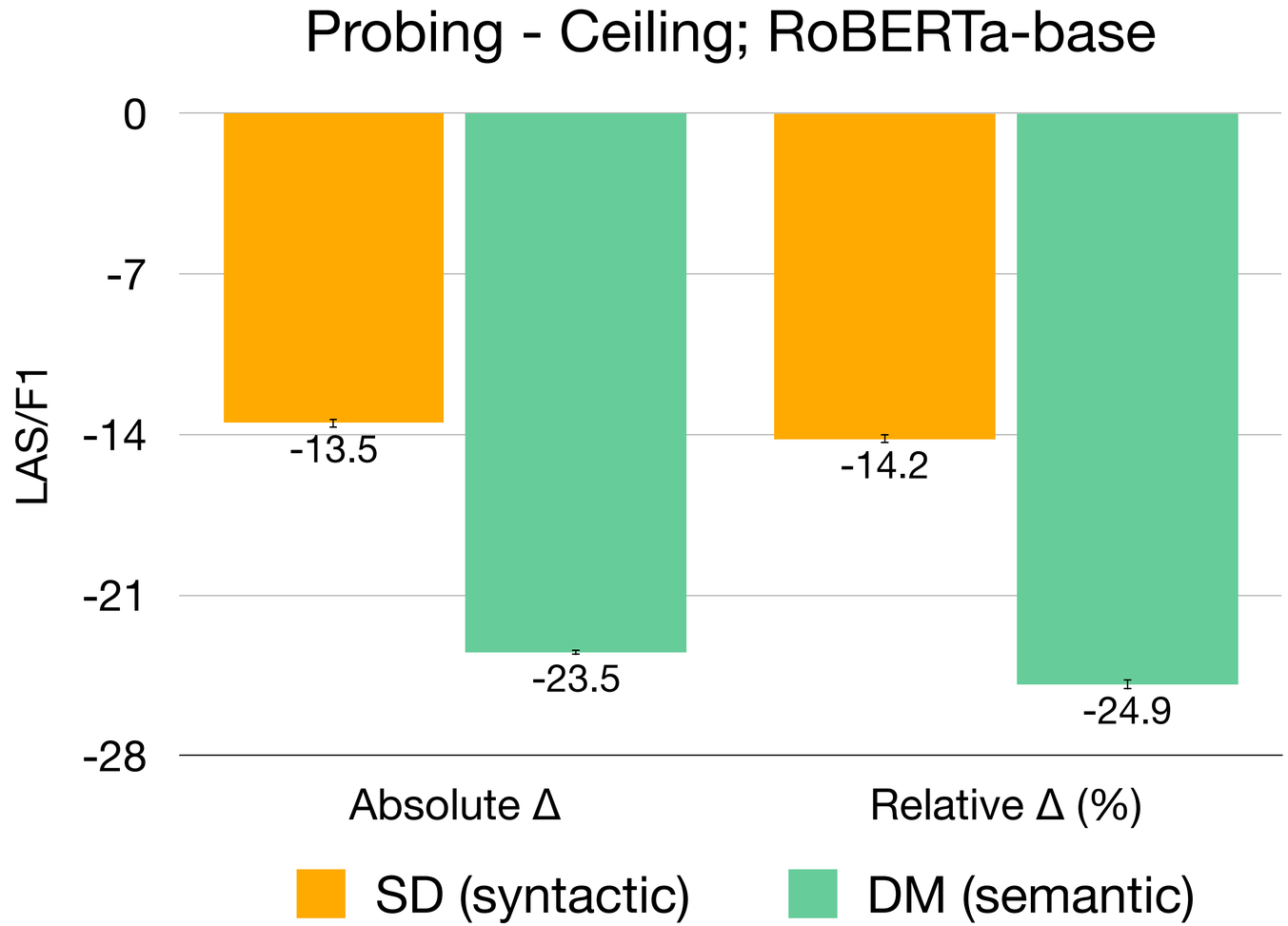
**Probing** model (Shi et al., 2016; Adi et al., 2017)




**This technique is impossible to adopt .**

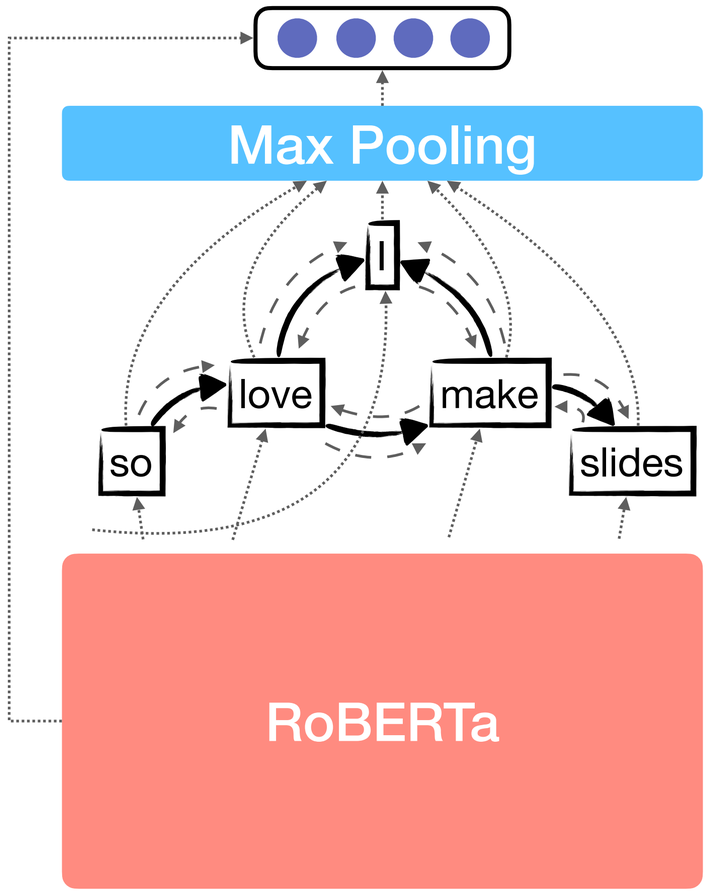
**Ceiling** model (Dozat and Manning, 2017, 2018)

# Probing RoBERTa with Semantics



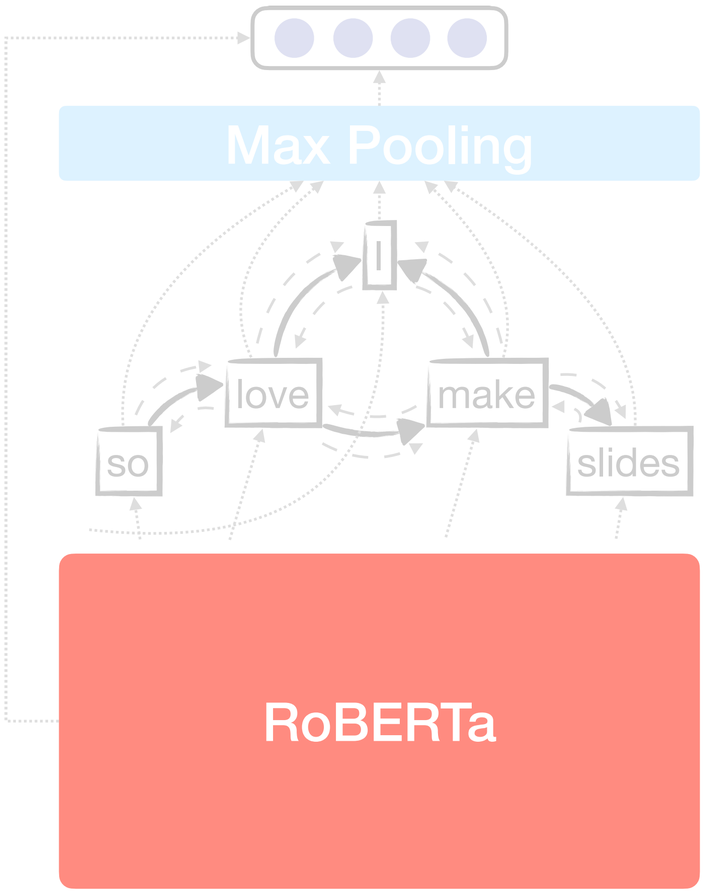
# Can we use semantics to augment pretrained transformers?

# Semantics-Infused Finetuning (SIFT)




**I so love to make slides .**

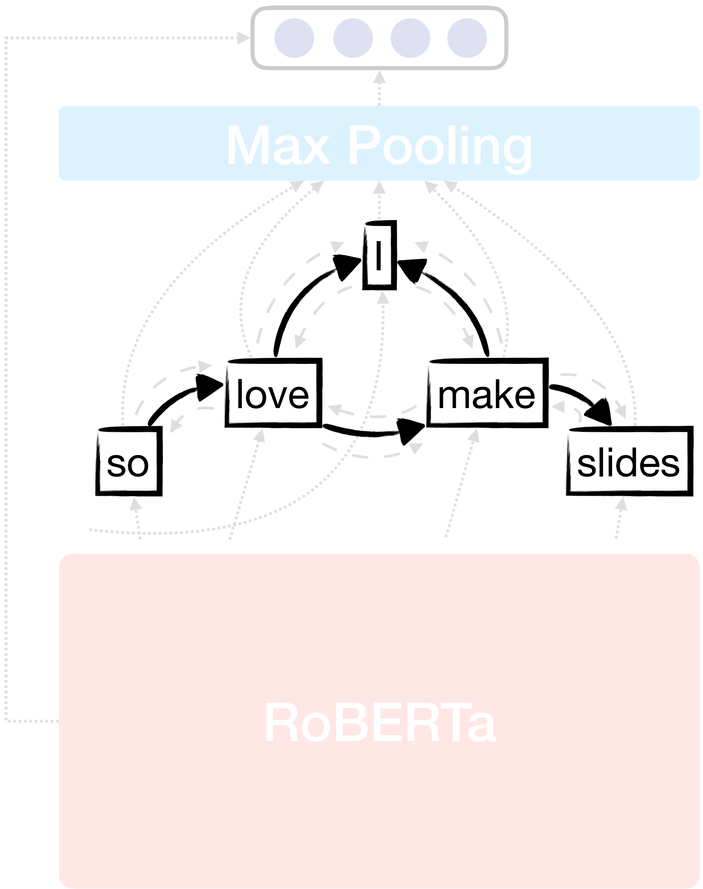
# Semantics-Infused Finetuning (SIFT)




**I so love to make slides .**

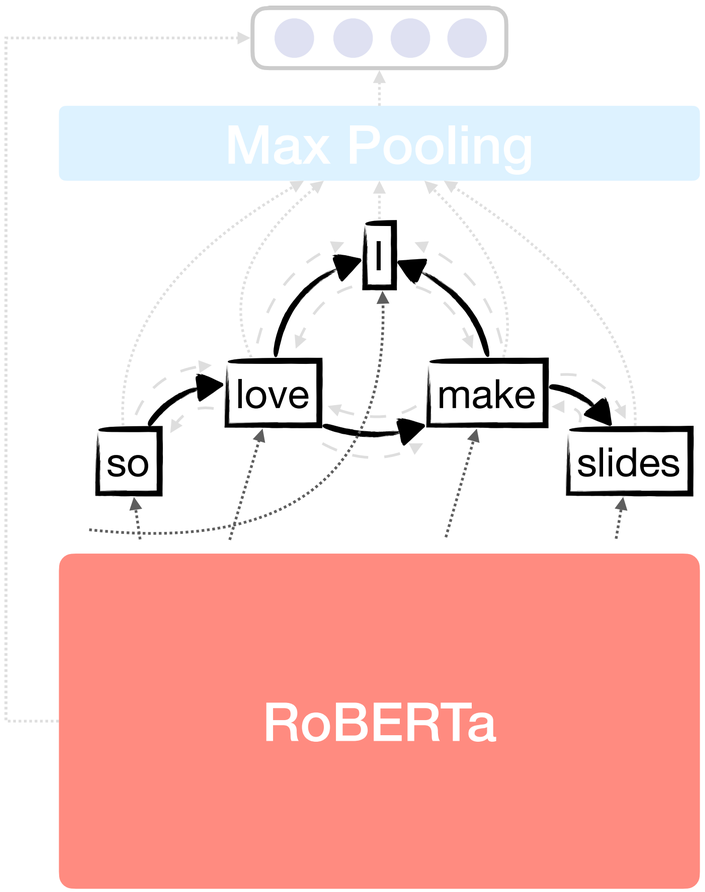
Che et al. (2019)

# Semantics-Infused Finetuning (SIFT)



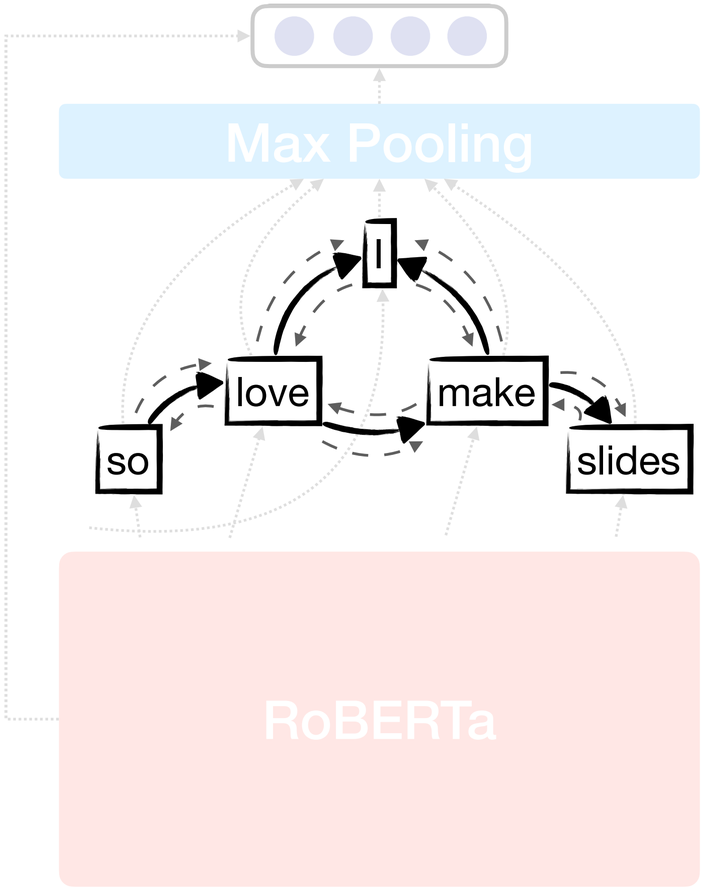

**I so love to make slides .**

# Semantics-Infused Finetuning (SIFT)

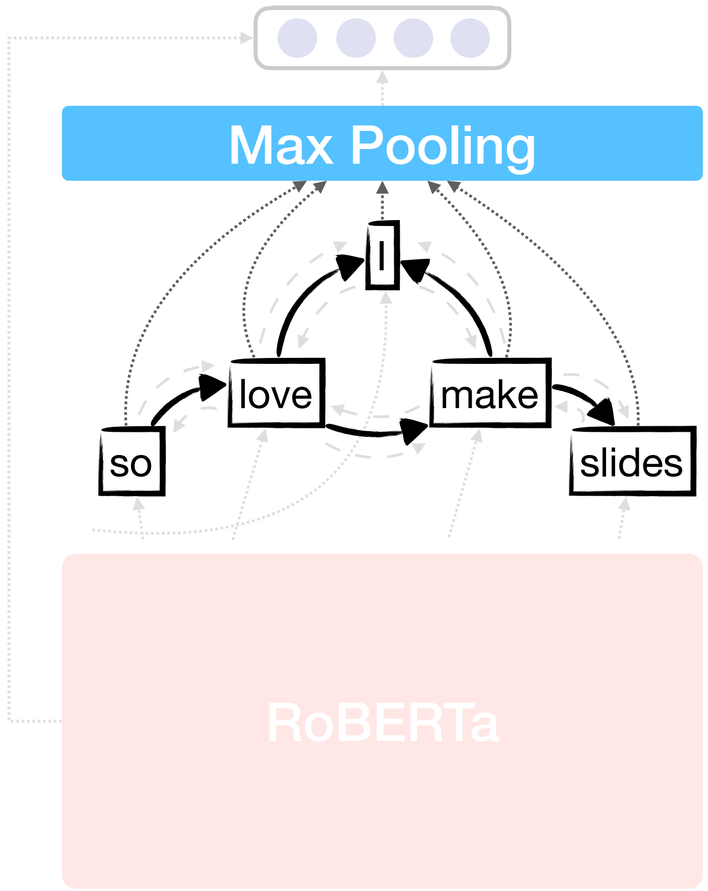


Schlichtkrull et al. (2017)

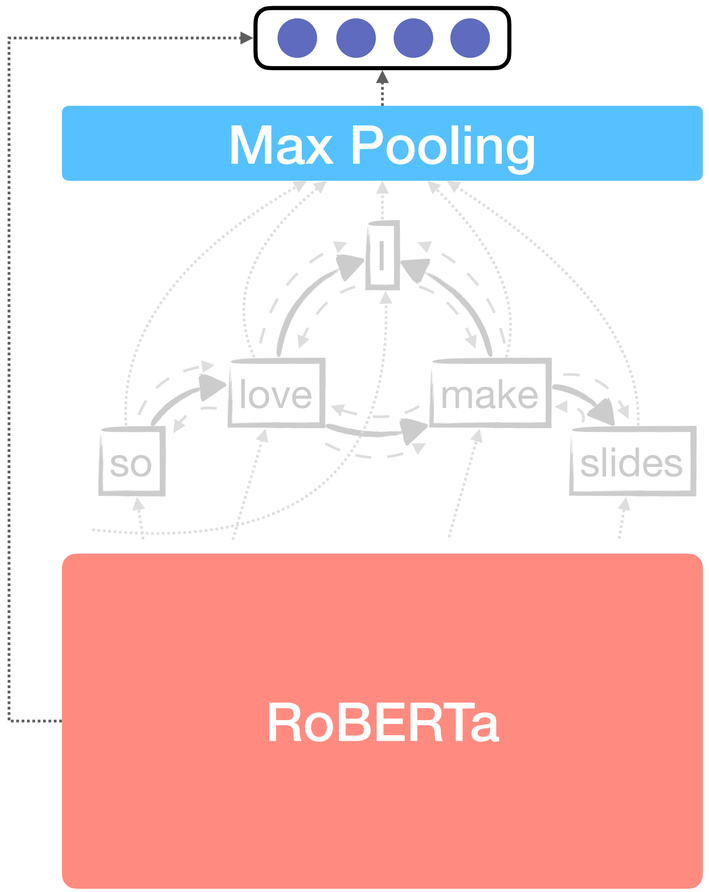
# Semantics-Infused Finetuning (SIFT)



# Semantics-Infused Finetuning (SIFT)



# Semantics-Infused Finetuning (SIFT)



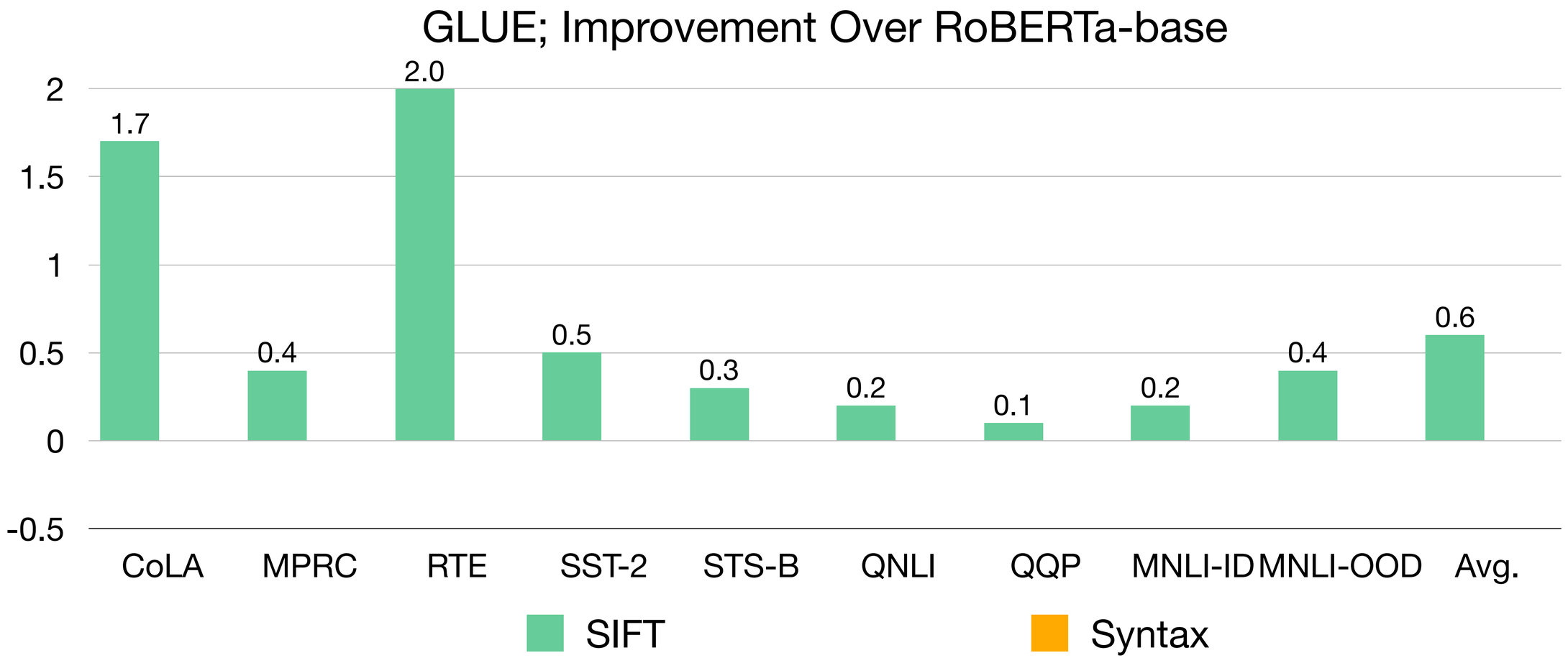

I so love to make slides .

# Experiments

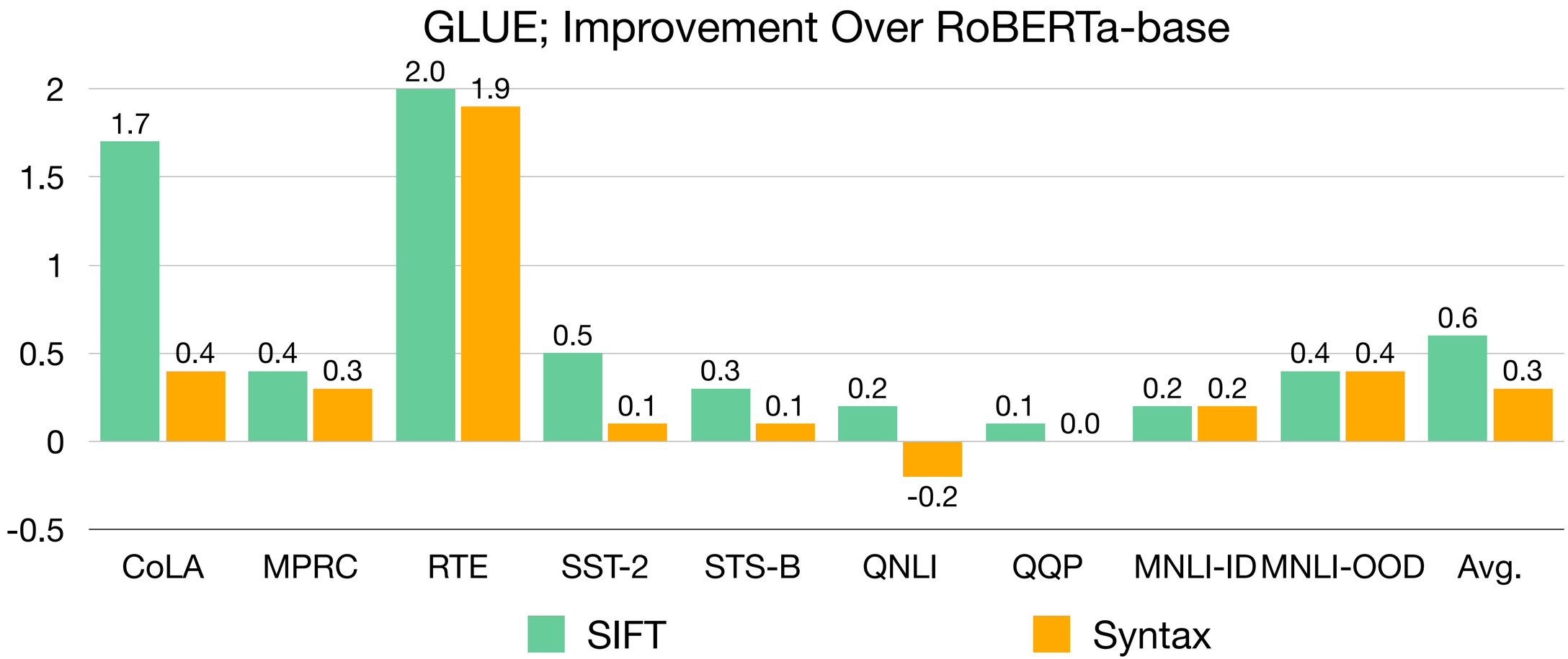
- Dataset: GLUE (Wang et al., 2018)
- Backbone: RoBERTa (Liu et al., 2019b)
- Parser: SOTA DM parser with 92.5 labeled F1 (Che et al., 2019)
- Graph Encoder: RGCN (Schlichtkrull et al., 2017)
  - 2 layers
  - Hidden dimension $\in \{256, 512, 768\}$
- Epochs $\in \{3, 10, 20\}$, learning rate $\in \{1 \times 10^{-4}, 2 \times 10^{-5}\}$

# Results

# Results


GLUE; Improvement Over RoBERTa-base
2
1.5
1
0.5
0
-0.5
1.7
0.4
2.0
0.5
0.3
0.2
0.1
0.2
0.4
0.6
CoLA
MPRC
RTE
SST-2
STS-B
QNLI
QQP
MNLI-ID
MNLI-OOD
Avg.
SIFT
Syntax

# Results


GLUE; Improvement Over RoBERTa-base
2
1.5
1
0.5
0
-0.5
1.7
0.4
0.4
0.3
2.0
1.9
0.5
0.1
0.3
0.1
0.2
-0.2
0.1
0.0
0.2
0.2
0.4
0.4
0.6
0.3
CoLA
MPRC
RTE
SST-2
STS-B
QNLI
QQP
MNLI-ID
MNLI-OOD
Avg.
SIFT
Syntax

# Analysis: When Do Semantic Structures Help?

- Two datasets
  - HANS tests if a model uses **invalid reasoning heuristics** (McCoy et al., 2019)
  - GLUE diagnostics tests the model capability in various **linguistic phenomena** (Wang et al., 2018)
- Examine a model trained on existing NLI datasets with synthetic NLI examples

# Analysis: HANS Lexical Overlap

**The actor stopped the banker.** does not entail **The banker stopped the actor.**

| RoBERTa | SIFT |
|---|---|
| 68.1 | **71.0** (+2.9) |

# Analysis: HANS Subsequence

**The judges heard the actor resigned.** does not entail **The judges heard the actor.**

| RoBERTa | SIFT |
|---|---|
| 25.8 | **29.5** (+3.7) |

# Analysis: HANS Constituent

**If the actor slept, the senator ran.** does not entail **The actor slept.**

| RoBERTa | SIFT |
|---|---|
| **37.9** | 37.6 (-0.3) |

# Analysis: HANS Constituent

**If the actor slept, the senator ran.** does not entail **The actor slept.**

**Before the actor slept, the senator ran.** entails

| RoBERTa | SIFT |
|---|---|
| **37.9** | 37.6 (-0.3) |

# Analysis: GLUE Diagnostics

| | | | | RoBERTa | SIFT |
|---|---|---|---|---|---|
| Pred-Arg Structure | I opened the door. | entails | The door opened. | 43.5 | **44.6** (+1.1) |
| | | does not entail | I opened. | | |
| Logic | I have no pet puppy. | entails | I have no corgi pet puppy. | 36.2 | **38.3** (+2.1) |
| | | does not entail | I have no pet. | | |

# Analysis: GLUE Diagnostics

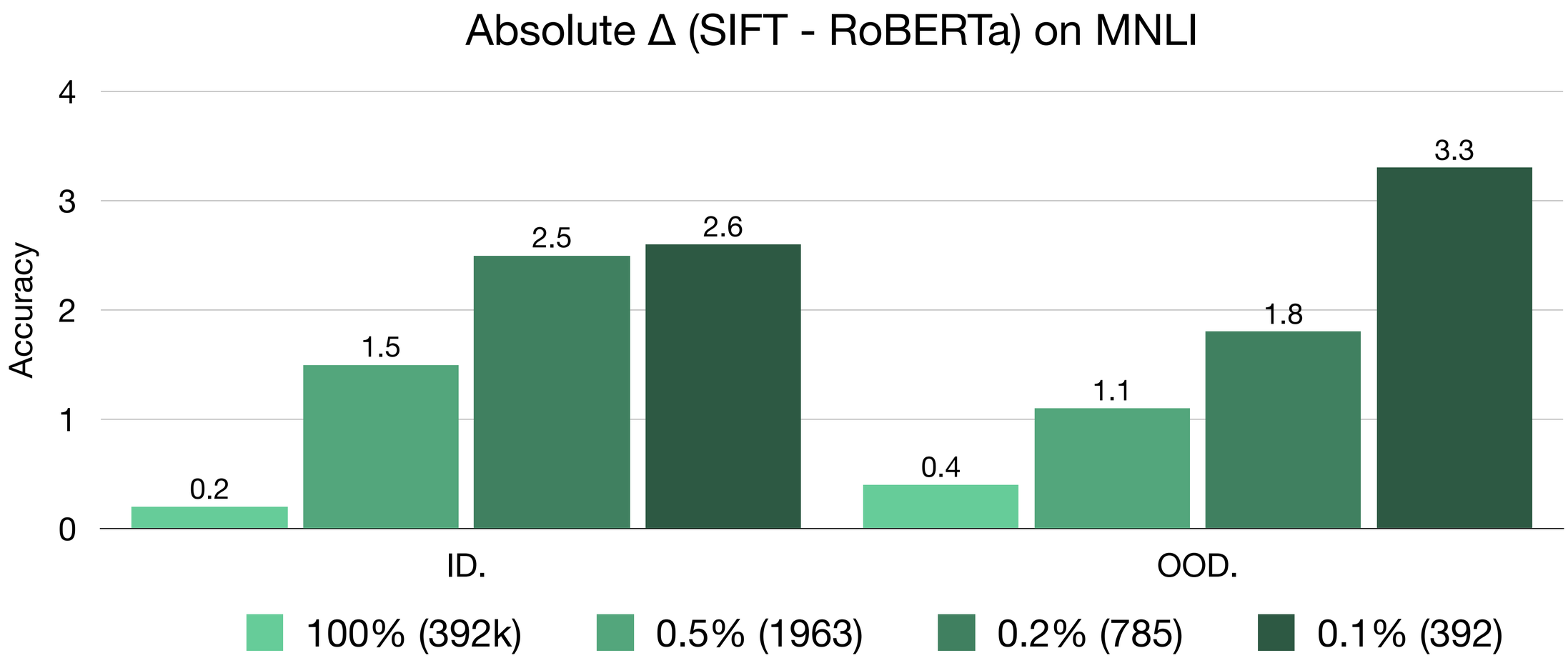
| | | | | RoBERTa | SIFT |
|---|---|---|---|---|---|
| Lexical Semantics | **I have a dog.** | entails | **I have an animal.** | **45.6** | 44.8 (-0.8) |
| | | does not entail | **I have a cat.** | | |
| Knowledge | **I live in Seattle.** | entails | **I live in the U.S.** | **28.0** | 26.3 (-1.7) |
| | | does not entail | **I live in Antarctica.** | | |

# Analysis: Sample Efficiency

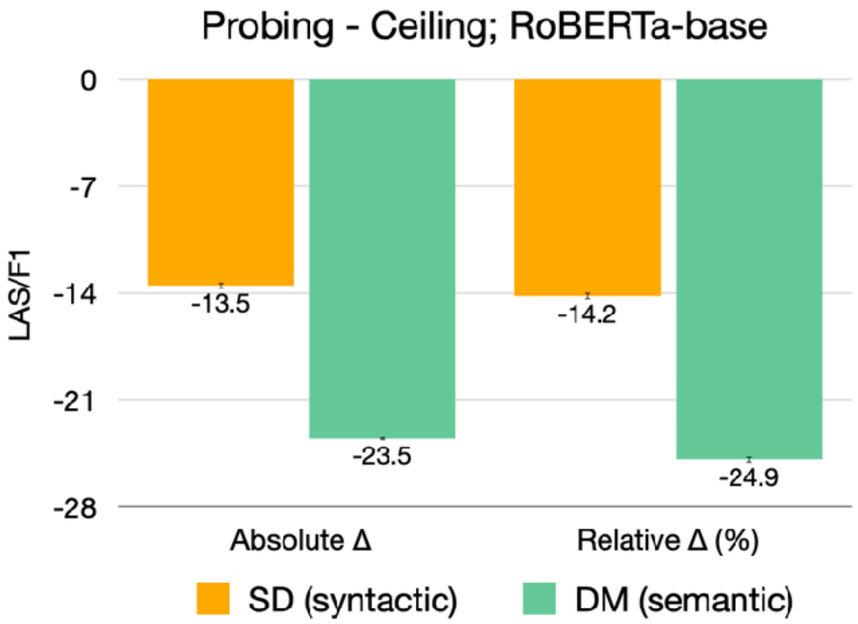
- Use the same downsampled MNLI training set to train RoBERTa & SIFT

# Analysis: Sample Efficiency

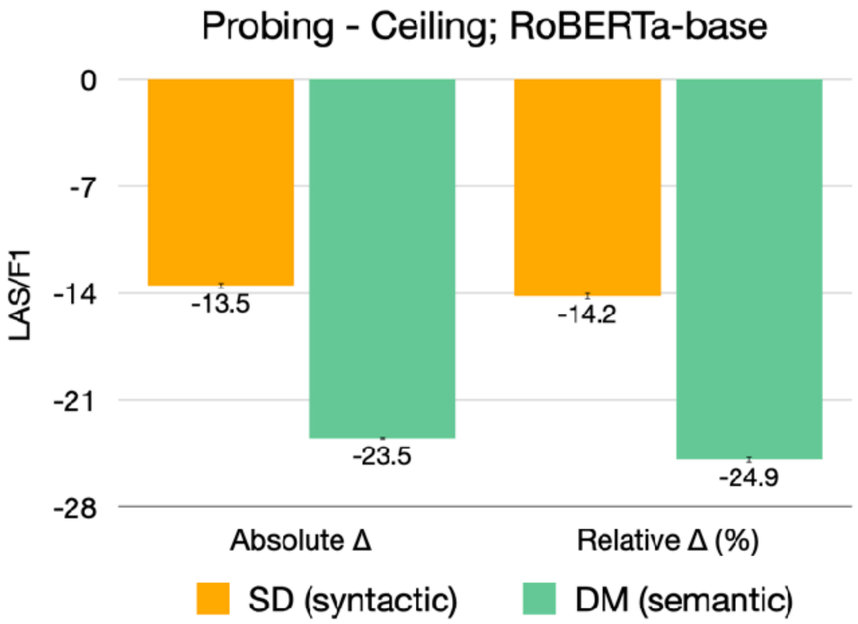
- Use the same downsampled MNLI training set to train RoBERTa & SIFT

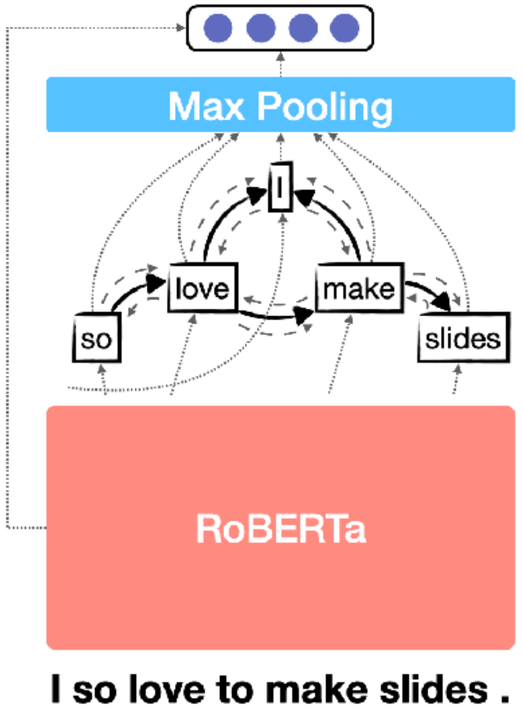


# Summary


Probing - Ceiling; RoBERTa-base
0
-7
-14
-21
-28
LAS/F1
-13.5
-23.5
-14.2
-24.9
Absolute Δ
Relative Δ (%)
SD (syntactic)
DM (semantic)

# Summary




**I so love to make slides .**

# Further Readings

- A Comprehensive Survey on Graph Neural Networks (Wu et al., 2019)
- Deep Learning on Graphs: A Survey (Zhang et al., 2020)
- Graph Neural Networks: A Review of Methods and Applications (Zhou et al., 2021)
- Frameworks
  - DGL: https://github.com/dmlc/dgl
  - PyTorch Geometric: https://github.com/rusty1s/pytorch_geometric

# Questions?

# References


Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient training and effective encoding. In Proc. of MRP.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2018. Neural Message Passing for Quantum Chemistry. In Proc. of ICML.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-Transformer. In Proc. of NAACL.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom?: A contrastive study of syntactosemantic dependencies. In Proc. of LAW.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In Proc. of ICLR.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The Second Shared Task on Cross-Framework and Cross-Lingual Meaning Representation Parsing. In Proc. of CoNLL.

# References


Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In Proc. of SemEval.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In Proc. of European Semantic Web Conference.

Petar Velickovič, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In Proc. of ICLR.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2019. A Comprehensive Survey on Graph Neural Networks. In IEEE Transactions on Neural Networks and Learning Systems.

Zhaofeng Wu, Hao Peng, and Noah A. Smith. 2021. Infusing Finetuning with Semantic Dependencies. In TACL.

Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep Learning on Graphs: A Survey. In IEEE Transactions on Knowledge and Data Engineering.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2021. Graph Neural Networks: A Review of Methods and Applications. In AI Open.