Deep Learning on Graphs for Natural Language Processing

Lingfei Wu, Yu Chen, Heng Ji, and Bang Liu

SIGIR-2021 Tutorial
July 11th, 2021
Outline

**DLG4NLP Introduction**
- Why Graphs for NLP?
- Conventional ML for NLP
- Deep Learning on Graphs: Foundations and Models

**DLG4NLP Foundations**
- Graph Construction for NLP
- Graph Representation Learning for NLP
- Graph Encoder-Decoder Models for NLP

**DLG4NLP Applications**
- Information Extraction
- Semantic Parsing and Machine Reading Comprehension
- Natural Language Generation and Machine Translation

**DLG4NLP Future Directions**

**DLG4NLP Demos**

**DLG4NLP Introduction, Foundations, Applications**
Introduction

DLG4NLP
Why graphs?

- Graphs are a general language for describing and modeling complex systems.
Graph-structured data are ubiquitous

Interne
Social networks
Information retrieval
Biomedical graphs
Program graphs
Scene graphs
Graphs are ubiquitous in NLP As Well

Dependency graph

AMR graph

Constituency graph

IE graph

SQL graph
Machine Learning on Graphs for NLP
Natural Language Processing: A Graph Perspective

• Represent natural language as a bag of tokens
  • BOW, TF-IDF
  • Topic Modeling: text as a mixture of topics

• Represent natural language as a sequence of tokens
  • Linear-chain CRF
  • Word2vec, Glove

• Represent natural language as a graph
  • Dependency graphs, constituency graphs, AMR graphs, IE graphs, and knowledge graphs
  • Text graph containing multiple hierarchies of elements, i.e. document, sentence and word
Graph Based Methods for NLP

• Random Walk Algorithms
  • Generate random paths, one can obtain a stationary distribution over all the nodes in a graph
  • Applications: semantic similarity of texts, name disambiguation

• Graph Clustering Algorithms
  • Spectral clustering, random walk clustering and min-cut clustering for text clustering

• Graph Matching Algorithms
  • Compute the similarity between two graphs for textual entailment task

• Label Propagation Algorithms
  • Propagate labels from labeled data points to previously unlabeled data points
  • Applications: word-sense disambiguation, sentiment analysis

[Mihalcea and Radev, 2011]
Deep Learning on Graphs: Foundations and Models
Machine Learning Lifecycle

• (Supervised) Machine Learning Lifecycle: feature learning is the key
Feature Learning in Graphs

• Our Goal: Design efficient task-independent/ task-dependent feature learning for machine learning in graphs!
Graph Neural Networks: Foundations

- Learning node embeddings:
  \[ h_i^{(l)} = f_{\text{filter}}(A, H^{(l-1)}) \]
  - A graph filter
  - adjacency matrix
  - Updated node embeddings
  - Input node embeddings

- Learning graph-level embeddings:
  \[ A', H' = f_{\text{pool}}(A, H) \]
  - A small graph w/ fewer nodes
  - Input graph
  - New node embeddings
  - Input node embeddings

- Spectral-based
- Spatial-based
- Attention-based
- Recurrent-based

- Flat Graph Pooling (i.e. Max, Ave, Min)
- Hierarchical Graph Pooling (i.e. Diffpool)
Graph Neural Networks: Basic Model

- **Key idea**: Generate node embeddings based on local neighborhoods.
Neighborhood Aggregation

• **Intuition:** Network neighborhood defines a computation graph!

Every node defines a unique computation graph!
Neighborhood Aggregation

- Nodes have embeddings at each layer.
- Model can be arbitrary depth.
- “layer-0” embedding of node $i$ is its input feature, i.e. $x_i$. 
Overview of GNN Model

1) Define a neighborhood aggregation function

2) Define a loss function on the embeddings, $L(z_v)$
Overview of GNN Model

3) Train on a set of nodes, i.e., a batch of computation graphs
Overview of GNN Model

4) Generate embeddings for nodes as needed

Even for nodes we never trained on!
GNN Model: A Case Study

- **Basic approach**: Average neighbor information and apply a neural network

  1) average messages from neighbors

  2) apply neural network
GNN Model: A Case Study

• Basic approach: Average neighbor information and apply a neural network.

\[ h^0_v = x_v \]
\[ h^k_v = \sigma \left( \sum_{u \in N(v)} \frac{h^{k-1}_u}{|N(v)|} + B_k h^{k-1}_v \right), \quad \forall k > 0 \]
GNN Model: Quick Summary

• Key idea: generate node embeddings by aggregating neighborhood information.
  ▪ Allows for parameter sharing in the encoder
  ▪ Allows for inductive learning
Graph Neural Networks: Popular Models

• Spectral-based Graph Filters
  • GCN (Kipf & Welling, ICLR 2017), Chebyshev-GNN (Defferrard et al. NIPS 2016)

• Spatial-based Graph Filters
  • MPNN (Gilmer et al. ICML 2017), GraphSage (Hamilton et al. NIPS 2017)
  • GIN (Xu et al. ICLR 2019)

• Attention-based Graph Filters
  • GAT (Velickovic et al. ICLR 2018)

• Recurrent-based Graph Filters
  • GGNN (Li et al. ICLR 2016)
Graph Convolution Networks (GCN)

Key idea: spectral convolution on graphs

\[ f_{\text{filter}} \ast x_i = U f(\Lambda) U^T x_i \]

\[ f'_{\text{filter}} \ast x_i \approx \sum_{p=0}^{P} \theta' p T_p(\tilde{L}) x_i \]

\[ f_{\text{filter}} \ast h_i^{(l)} \approx \theta(I_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) h_i^{(l)} \]

\[ H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \]

Eigen-decomposition is expensive

Chebyshev polynomials accelerates but still not powerful

First-order approximation fast and powerful

Renormalization trick stabilizes the numerical computation

GCN in NLP Tasks:
- Text classification
- Question Answering
- Text Matching
- Topic Modeling
- Information Extraction
Message Passing Neural Network (MPNN)

**Key idea:** graph convolutions as a message passing process

MPNN:
\[
    h_i^{(l)} = f_{\text{filter}}(A, H^{(l-1)}) = f_U(h_i^{(l-1)}, \sum_{v_j \in N(v_i)} f_M(h_i^{(l-1)}, h_j^{(l-1)}, e_{i,j}))
\]

Expensive if the number of nodes are large

GraphSage:
\[
    f_{\text{filter}}(A, H^{(l-1)}) = \sigma(W^{(l)} \cdot f_M(h_i^{(l-1)}, \{h_j^{(l-1)}, \forall v_j \in N(v_i)\}))
\]

Sampling to obtain a fixed number of neighbors

MPNN and GraphSage in NLP Tasks:
- Knowledge graph
- Information extraction
- Semantic parsing
Graph Attention Network (GAT)

**Key idea:** dynamically learn the weights (attention scores) on the edges when performing message passing

\[
h_i^{(l)} = f_{\text{filter}}(A, H^{(l-1)}) = \sigma \left( \sum_{v_j \in N(v_i)} \alpha_{ij} W^{(l)} h_j^{(l-1)} \right)
\]

\[
\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(u^{(l)}^T [W^{(l)} h_i^{(l-1)} || W^{(l)} h_j^{(l-1)}]))}{\sum_{v_k \in N(v_i)} \exp(\text{LeakyReLU}(u^{(l)}^T [W^{(l)} h_i^{(l-1)} || W^{(l)} h_k^{(l-1)}]))}
\]

\[
f_{\text{filter}}(A, H^{(l-1)}) = ||_{k=1}^K \sigma \left( \sum_{v_j \in N(v_i)} \alpha_{ij}^k W_k^{(l)} h_j^{(l-1)} \right)
\]

\[
f_{\text{filter}}(A, H^{(L-1)}) = \sigma \left( \frac{1}{K} \sum_{k=1}^K \sum_{v_j \in N(v_i)} \alpha_{ij}^k W_k^{(L)} h_j^{(L-1)} \right)
\]

**GAT in NLP Tasks:**
- Text classification
- Question Answering
- Knowledge graph
- Information extraction
- Semantic parsing
Gated Graph Neural Networks (GGNN)

Key idea: the use of Gated Recurrent Units while taking into account edge type and directions

Zero-padding input node embeddings

Incoming & outcoming edges for node $v_i$

$$h_i^{(0)} = [x_i^T, 0]^T$$
$$a_i^{(l)} = A_i^T[h_1^{(l-1)} ... h_n^{(l-1)}]^T$$
$$h_i^{(l)} = \text{GRU}(a_i^{(l)}, h_i^{(l-1)})$$

GGNN in NLP Tasks:
- Semantic parsing
- Machine translation
DLG4NLP
Foundations
Graph Construction for NLP
Why Graph Construction for NLP?

- Representation power: graph > sequence > bag
- Different NLP tasks require different aspects of text, e.g., syntax, semantics.
- Different graphs capture different aspects of the text
- Two categories: static vs dynamic graph construction
- Goal: good downstream task performance

```
Text
```

convert to graph

many more graph options...
Static Graph Construction

• Problem setting:
  • **Input**: raw text (e.g., sentence, paragraph, document, corpus)
  • **Output**: graph

• Conducted during **preprocessing** by augmenting text with domain knowledge
Text input: are there ada jobs outside austin

Add additional sequential edges to
1) reserve sequential information in raw text
2) connect multiple dependency graphs in a paragraph
Static Graph Construction: Constituency Graph

Text input: are there ada jobs outside austin

Constituency parsing

Again, add additional sequential edges
Text input: Paul’s description of himself: a fighter
**Text input:** Paul, a renowned computer scientist, grew up in Seattle. He attended Lakeside School.

**Static Graph Construction: IE Graph**

- **OpenIE**
- **Coreference**

- **Paul**
- **He**
- **a renowned ...**

- **grew up in**
- **Seattle**

- **attended**
- **Lakeside School**
Static Graph Construction: Knowledge Graph

Get the concept sub-graph from KB

**Question:** who acted in the movies directed by the director of [Some Mother's Son]

**Answer:** Don Cheadle, Joaquin Phoenix
Static Graph Construction: Topic Graph

There was the $5 million Deutsche Bank Championship to prepare for and the Ryder Cup is a few weeks away, but the first order of business for Jim Furyk yesterday was to make sure his wife and children were headed for safety.

A sports psychologist says how footballers should prepare themselves for the high-pressure penalties.

Dolphin groups, or "pods", rely on socialites to keep them from collapsing, scientists claim.
Static Graph Construction: Similarity Graph

qpr keeper day heads for preston

former ni minister scott dies

uk will stand firm on eu rebate

Cranes: Flying giant returning to Ireland after 300 years

souness backs smith for scotland
Static Graph Construction: Co-occurrence Graph

Text input: To be, or not to be: ...

Co-occurrence matrix

<table>
<thead>
<tr>
<th></th>
<th>to</th>
<th>be</th>
<th>or</th>
<th>not</th>
</tr>
</thead>
<tbody>
<tr>
<td>to</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>be</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>not</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Co-occurrence graph
Static Graph Construction: SQL Graph

SQL query input: SELECT company WHERE assets > val₀ AND sales > val₀ AND industry_rank ≤ val₁
Static Graph Construction: Application-driven Graph

Question: Who is the director of the 2003 film which has scenes in it filmed at the Quality Cafe in Los Angeles?

1-hop

Quality Cafe (jazz club)

Quality Cafe was a historical restaurant and jazz club...

Old School (film)

Old School is a 2003 American comedy film... directed by Todd Phillips.

2-hop

Gone in 60 Seconds

Gone in 60 Seconds is a 2000 American action heist film... directed by Dominic Sena.

3-hop

Todd Phillips

Correct answer

Dominic Sena

Ming Ding et al. “Cognitive Graph for Multi-Hop Reading Comprehension at Scale”. ACL 2019.
Static Graph Construction: Summary

Widely used in various NLP applications such as NLG, MRC, semantic parsing, etc.
Dynamic Graph Construction

• Problem setting:
  • Input: raw text (e.g., sentence, paragraph, document, corpus)
  • Output: graph

• Graph structure (adjacency matrix) learning on the fly, joint with graph representation learning
Dynamic Graph Construction: Overview

Data points (e.g., words, sentences, documents) → Graph similarity

Graph similarity metric learning → Fully-connected weighted graph

Graph sparsification → Learned graph

Learned graph → GNN → y

Combining intrinsic and implicit graph structures
Dynamic Graph Construction Outline

- Graph Similarity Metric Learning Techniques
- Graph Sparsification Techniques
- Combining Intrinsic Graph Structures and Implicit Graph Structures
- Learning Paradigms
Graph Similarity Metric Learning Techniques

- Graph structure learning as *similarity metric learning* (in the node embedding space)
- Enabling *inductive learning*
- Various metric functions

- Node Embedding Based Similarity Metric Learning
- Structure-aware Similarity Metric Learning
- Attention-based Similarity Metric Functions
- Cosine-based Similarity Metric Functions
- Structure-aware Attention Mechanism
Node Embedding Based Similarity Metric Learning

• Learning a weighted adjacency matrix by computing the pair-wise node similarity in the embedding space

• Common metrics functions
  • Attention-based similarity metric functions
  • Cosine-based similarity metric functions

Data points (e.g., words, sentences, documents) → Learning pair-wise node similarity → Fully-connected weighted graph
Attention-based Similarity Metric Functions

Variant 1)

\[ S_{i,j} = \left( v_i \odot u \right)^T v_j \]

Node feature vector
Non-negative learnable weight vector

Variant 2)

\[ S_{i,j} = \text{ReLU}(Wv_i)^T \text{ReLU}(Wv_j) \]

Learnable weight matrix

Data points (e.g., words, sentences, documents)

Fully-connected weighted graph


Cosine-based Similarity Metric Functions

\[ S_{i,j}^p = \cos(w_p \odot v_i, w_p \odot v_j) \]

Learnable weight vector

\[ S_{i,j} = \frac{1}{m} \sum_{p=1}^{m} S_{i,j}^p \]

Multi-head similarity scores

Data points (e.g., words, sentences, documents)

Fully-connected weighted graph

Structure-aware Similarity Metric Learning

• Learning a weighted adjacency matrix by computing the pair-wise node similarity in the embedding space

• Considering existing edge information of the intrinsic graph in addition to the node information
Attention-based Similarity Metric Functions

Variant 1)

\[
S_{i,j}^l = \text{softmax}(u^T \tanh(W[h_i^l, h_j^l, v_i, v_j, e_{i,j}]))
\]

Variant 2)

\[
S_{i,j} = \frac{\text{ReLU}(W^Q v_i)^T (\text{ReLU}(W^K v_i) + \text{ReLU}(W^R e_{i,j}))}{\sqrt{d}}
\]


Graph Sparsification Techniques

• Similarity metric functions learn a fully-connected graph
• Fully-connected graph is computationally expensive and might introduce noise
• Enforcing sparsity to the learned graph structure
• Various techniques
Common Graph Sparsification Options

Option 1) KNN-style Sparsification

$$A_{i,:} = \text{topk}(S_{i,:})$$

Option 2) epsilon-neighborhood Sparsification

$$A_{i,j} = \begin{cases} S_{i,j} & S_{i,j} > \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

Option 3) graph Regularization

$$\frac{1}{n^2} \| A \|_F^2$$
Combining Intrinsic and Implicit Graph Structures

- Intrinsic graph typically still carries rich and useful information
- Learned implicit graph is potentially a “shift” (e.g., substructures) from the intrinsic graph structure

\[
\tilde{A} = \lambda L^{(0)} + (1 - \lambda)f(A)
\]

Normalized graph Laplacian

f(A) can be arbitrary operation, e.g., graph Laplacian, row-normalization


Learning Paradigms: Joint Learning

Node features & (optional) initial graph structure

Learned graph structure

Downstream task prediction

Graph Learner

GraphFlow

GNN

Learning Paradigms: Adaptive Learning

Node features & (optional) initial graph structure

Repeat for fixed num. of stacked GNN layers

Downstream task prediction

Learning Paradigms: Iterative Learning

Node features & (optional) initial graph structure

Downstream task prediction

Graph Learner

Learned graph structure

GNN

Node embeddings

Repeat until condition satisfied

Dynamic Graph Construction Summary

- **Graph Similarity Metric Learning Techniques**
  - Node Embedding Based Similarity Metric Learning
  - Structure-aware Similarity Metric Learning

- **Graph Sparsification Techniques**
  - KNN-style Sparsification
  - Epsilon-neighborhood Sparsification
  - Graph Regularization

- **Combining Intrinsic Graph Structures and Implicit Graph Structures**

- **Learning Paradigms**
  - Joint Learning of Graph Structures and Representations
  - Adaptive Learning of Graph Structures and Representations
  - Iterative Learning of Graph Structures and Representations
## Static vs. Dynamic Graph Construction

<table>
<thead>
<tr>
<th>Static graph construction</th>
<th>Dynamic graph construction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros</strong></td>
<td><strong>Pros</strong></td>
</tr>
<tr>
<td>prior knowledge</td>
<td>no domain expertise</td>
</tr>
<tr>
<td></td>
<td>joint graph structure &amp; representation learning</td>
</tr>
<tr>
<td><strong>Cons</strong></td>
<td><strong>Cons</strong></td>
</tr>
<tr>
<td>extensive domain expertise</td>
<td>scalability</td>
</tr>
<tr>
<td>• error-prone (e.g., noisy, incomplete)</td>
<td>explainability</td>
</tr>
<tr>
<td>• sub-optimal</td>
<td></td>
</tr>
<tr>
<td>• disjoint graph structure &amp; representation learning</td>
<td></td>
</tr>
<tr>
<td>• error accumulation</td>
<td></td>
</tr>
</tbody>
</table>

New topic in DLG4NLP!
Static vs. Dynamic Graph Construction (cont)

When to use static graph construction

- Domain knowledge which fits the task and can be presented as a graph

When to use dynamic graph construction

- Lack of domain knowledge which fits the task or can be presented as a graph
- Domain knowledge is incomplete or might contain noise
- To learn implicit graph which augments the static graph

Graph Representation Learning for NLP
GNNs for Graph Representation Learning
Homogeneous vs Multi-relational vs Heterogeneous Graphs

<table>
<thead>
<tr>
<th>Graph types</th>
<th>Homogeneous</th>
<th>Multi-relational</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td># of node types</td>
<td>1</td>
<td>1</td>
<td>&gt; 1</td>
</tr>
<tr>
<td># of edge types</td>
<td>1</td>
<td>&gt; 1</td>
<td>&gt;= 1</td>
</tr>
</tbody>
</table>
Which GNNs to Use Given a Graph?

Graph

Homogeneous graph?

YES

NO

Homogeneous GNNs

GCN, GAT, GraphSAGE, GGNN, ...

NO

Convert to homogeneous graph?

YES

NO

Multi-relational GNNs

GAT, GGNN, ...

Bidirectional?

YES

NO

Heterogeneous GNNs

Single node type?

YES

NO

NO (directed edges only)

NO (edge directions as types)

Graph embeddings

NO

Undirected graph?

YES

NO
Homogeneous GNNs for NLP

• When to use homogeneous GNNs?

• Homogeneous GNNs
  • GCN
  • GAT
  • GraphSAGE
  • GGNN
  • ...

Graph

Homogeneous
graph?

Ignore node/edge
types, Levi graph, …

Convert to
homogeneous
graph?

Homogeneous
GNNs

Graph
embeddings
Non-homogeneous to Homogeneous Conversion via Levi Graph

Levi graph conversion

Levi graph: edges as new nodes
How to Handle Edge Direction Information?

• Edge direction is important (think about BiLSTM, BERT)

• Common strategies for handling directed graphs
  a) Message passing only along directed edges (e.g., GAT, GGNN)
  b) Regarding edge directions as edge types (i.e., adding “reverse” edges)
  c) Bidirectional GNNs
Edge Directions as Edge Types

• Regarding edge directions as edge types, resulting in a multi-relational graph

\[
\text{dir}_{i,j} = \begin{cases} 
\text{default}, & e_{i,j} \text{ is originally existing in the graph} \\
\text{inverse}, & e_{i,j} \text{ is the inverse edge} \\
\text{self}, & i = j 
\end{cases}
\]

Then we can apply multi-relational GNNs
Bidirectional GNNs for Directed Graphs

Bi-Sep GNNs formulation:

Run multi-hop backward/forward GNN on the graph

\[
\begin{align*}
\mathbf{h}^k_{i,-} &= GNN(\mathbf{h}^{k-1}_{i,-}, \{\mathbf{h}^{k-1}_{j,-} : \forall v_j \in \mathcal{N}_-(v_i)\}) \\
\mathbf{h}^k_{i,=} &= GNN(\mathbf{h}^{k-1}_{i,=}, \{\mathbf{h}^{k-1}_{j,=} : \forall v_j \in \mathcal{N}_+(v_i)\})
\end{align*}
\]

Concatenate backward/forward node embeddings at last hop

\[
\mathbf{h}^K_i = \mathbf{h}^K_{i,-} \mid\mid \mathbf{h}^K_{i,=}
\]

Xu et al. “Graph2Seq: Graph to Sequence Learning with Attention-based Neural Networks”. 2018.
Bidirectional GNNs for Directed Graphs (cont)

Bi-Fuse GNNs formulation:

Run one-hop backward/forward node aggregation

\[
\mathbf{h}^k_{\mathcal{N}_-}(v_i) = AGG(\mathbf{h}^{k-1}_i, \{ \mathbf{h}^{k-1}_j : \forall v_j \in \mathcal{N}_-(v_i) \})
\]

\[
\mathbf{h}^k_{\mathcal{N}_+}(v_i) = AGG(\mathbf{h}^{k-1}_i, \{ \mathbf{h}^{k-1}_j : \forall v_j \in \mathcal{N}_+(v_i) \})
\]

Fuse backward/forward aggregation vectors at each hop

\[
\mathbf{h}^k_{\mathcal{N}}(v_i) = Fuse(\mathbf{h}^k_{\mathcal{N}_-}(v_i), \mathbf{h}^k_{\mathcal{N}_+}(v_i))
\]

Update node embeddings with fused aggregation vectors at each hop

\[
\mathbf{h}^k_i = \sigma(\mathbf{h}^{k-1}_i, \mathbf{h}^k_{\mathcal{N}}(v_i))
\]

---

Multi-relational GNNs for NLP

• When to use multi-relational GNNs?

• Multi-relational GNNs
  a) Including relation-specific transformation parameters in GNN
  b) Including edge embeddings in GNN
  c) Multi-relational Graph Transformers
R-GNN: Overview

\[ h_i^k = \sigma(h_i^{k-1}, \sum_{v_j \in \mathcal{N}(v_i)} AGG(h_j^{k-1}, \theta^k)) \]

GNN

1) relation-specific transformation, e.g., node feature transformation, attention weight ...

\[ h_i^k = \sigma(h_i^{k-1}, \sum_{r \in \mathcal{E}} \sum_{v_j \in \mathcal{N}_r(v_i)} AGG(h_j^{k-1}, \theta_r^k)) \]

2) aggregation per relation-specific subgraph

AGG

rel_1

rel_2

... rel_N

σ
R-GNN Variant: R-GCN

- Relation-specific node feature transformation during neighborhood aggregation

\[ h^k_i = \sigma \left( \sum_{r \in \mathcal{E}} \sum_{v_j \in \mathcal{N}_r(v_i)} \frac{1}{c_{i,r}} \left( W^k_r h^k_{j} + W^k_0 h^k_{i} \right) \right), \quad c_{i,r} = |\mathcal{N}_r(v_i)| \]

Relation-specific d x d learnable weight matrix

R-GNN: Avoiding Over-parameterization

Learning $d \times d$ transformation weight matrix for each relation is expensive! $O(Rd^2)$ parameters every GNN layer where $R$ is the num of relation types.

How to avoid over-parameterization?

Option 1) basis decomposition - linear hypothesis

$$\theta_{rk} = \sum_{b=1}^{B} a_{rb}^{k} \mathbf{V}_b^{(k)}, \quad \mathbf{V}_b^{(k)} \in \mathbb{R}^{d \times d} \quad \text{O}(RB + Bd^2) \text{ parameters}$$

Basis matrices

Option 2) block-diagonal decomposition - sparsity hypothesis

$$\theta_{rk} = \bigoplus_{b=1}^{B} \mathbf{Q}_{br}^{k} = \text{diag}(\mathbf{Q}_{1r}^{k}, \mathbf{Q}_{2r}^{k}, \ldots, \mathbf{Q}_{Br}^{k}), \quad \mathbf{Q}_{br}^{(k)} \in \mathbb{R}^{d/B \times d/B} \quad \text{O}(Rd^2/B) \text{ parameters}$$

Submatrices
Including Edge Embeddings in GNNs

Variant 1) Include edge embeddings in message passing

\[ h^k_i = \sigma(h^{k-1}_i, \sum_{v_j \in \mathcal{N}(v_i)} AGG(h^{k-1}_j, e_{i,j}, \theta^k)) \]

Variant 2) Update edge embedding in message passing

\[ h^k_i = \sigma(h^{k-1}_i, \sum_{v_j \in \mathcal{N}(v_i)} AGG(h^{k-1}_j, e_{i,j}^{k-1}, \theta^k)), \quad e^k_{i,j} = f(e^{k-1}_{i,j}, \theta_{rel}) \]


Multi-relational Graph Transformers

• Transformers as a special class of GNNs which
  • jointly learn and encode a fully-connected graph via self-attention
  • share many similarities with GAT
  • fail to effectively handle arbitrary graph-structured data
    • e.g., position embeddings for sequential data, removing position embeddings for set

• Multi-relational graph transformers
  • employed with structure-aware self-attention
  • respect various relation types
R-GAT based Graph Transformers

GAT-like masked attention

\[
z_{i}^{r,k} = \sum_{v_{j} \in \mathcal{N}_{r}(v_{i})} (\alpha_{i,j}^{k}) W_{V}^{k} h_{j}^{k-1}, r \in \mathcal{E}
\]

Relation-specific learnable weight matrix

\[
h_{i}^{k} = FFN_{k}^{k}(W_{O}^{k}[z_{i}^{R_{1},k}, ..., z_{i}^{R_{m},k}])
\]

Yao et al. “Heterogeneous Graph Transformer for Graph-to-Sequence Learning”. ACL 2020.
Structure-aware Self-attention based Graph Transformers

\[
\begin{align*}
    h_i^k &= \sum_j \alpha_{i,j}^k (W^k_v h_j^{k-1} + W^k_F e_{i,j}) \\
    \alpha_{i,j}^k &= \text{softmax}(u_{i,j}^k) \\
    u_{i,j}^k &= \frac{(W^k_Q h_{i}^{k-1})^T (W^k_K h_{j}^{k-1} + W^k_R e_{i,j})}{\sqrt{d}}
\end{align*}
\]

Heterogeneous GNNs

• When to use Heterogeneous GNNs?
• Heterogeneous GNNs
  a) Meta-path based Heterogeneous GNNs

Meta paths among author nodes
Meta-path based Heterogeneous GNN example: HAN

Step 1) type-specific node feature transformation

\[ h_i = W_{\tau(v_i)} v_i \]

Node-type specific learnable weight matrix

Step 2) node-level aggregation along each meta path

\[ z_{i,\Phi_k} = \sigma \left( \sum_{v_j \in N_{\Phi_k}(v_i)} \alpha_{i,j}^{\Phi_k} h_j \right) \]

Aggregate over neighboring nodes in k-length meta path

Step 3) meta-path level aggregation

\[ z_i = \sum_{k=1}^{p} \beta_{\Phi_k} z_{i,\Phi_k} \]

Attention weights over meta paths

---

Graph Encoder-Decoder Models for NLP
Seq2Seq: Applications and Challenges

- **Applications**
  - Machine translation
  - Natural language generation
  - Logic form translation
  - Information extraction

- **Challenges**
  - Only applied to problems whose inputs are represented as sequences
  - Cannot handle more complex structure such as graphs
  - Converting graph inputs into sequences inputs lose information
  - Augmenting original sequence inputs with additional structural information enhances word sequence feature
Graph-to-Sequence Model

[1] Kun Xu*, Lingfei Wu*, Zhiguo Wang, Yansong Feng, Michael Witbrock, and Vadim Sheinin (Equally Contributed), "Graph2Seq: Graph to Sequence Learning with Attention-based Neural Networks", arXiv 2018.

Bidirectional GNNs for Directed Graphs

Bi-Sep GNNs formulation:

Run multi-hop backward/forward GNN on the graph

\[ h_{i,-}^k = GNN(h_{i,-}^{k-1}, \{ h_{j,-}^{k-1} : \forall v_j \in N_-(v_i) \} ) \]

\[ h_{i,-}^k = GNN(h_{i,-}^{k-1}, \{ h_{j,-}^{k-1} : \forall v_j \in N_-(v_i) \} ) \]

Concatenate backward/forward node embeddings at last hop

\[ h_i^K = h_{i,-}^K || h_{i,-}^K \]

Xu et al. “Graph2Seq: Graph to Sequence Learning with Attention-based Neural Networks”. 2018.
Bidirectional GNNs for Directed Graphs (cont)

Bi-Fuse GNNs formulation:

Run one-hop backward/forward node aggregation

\[ h^k_{\mathcal{N}_-(v_i)} = AGG(h^k_{i}^{-1}, \{h^k_{j}^{-1} : \forall v_j \in \mathcal{N}_-(v_i) \}) \]
\[ h^k_{\mathcal{N}_+(v_i)} = AGG(h^k_{i}^{-1}, \{h^k_{j}^{-1} : \forall v_j \in \mathcal{N}_+(v_i) \}) \]

Fuse backward/forward aggregation vectors at each hop

\[ h^k_{\mathcal{N}(v_i)} = Fuse(h^k_{\mathcal{N}_-(v_i)}, h^k_{\mathcal{N}_+(v_i)}) \]

Update node embeddings with fused aggregation vectors at each hop

\[ h^k_{i} = \sigma(h^k_{i}^{-1}, h^k_{\mathcal{N}(v_i)}) \]

Graph Encoding

- **Graph embedding**
  - Pooling based graph embedding *(max, min and average pooling)*
  - Node based graph embedding
    - Add one super node which is connected to all other nodes in the graph
    - The embedding of this super node is treated as graph embedding
Attention Based Sequence Decoding

\[ c_i = \sum_{j=1}^{\mathcal{V}} \alpha_{ij} h_j, \text{ where } \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{\mathcal{V}} \exp(e_{ik})}, \quad e_{ij} = a(s_{i-1}, h_j) \]

context vector  node representation
Attention Based Sequence Decoding

\[ c_i = \sum_{j=1}^{\mathcal{V}} \alpha_{ij} h_j, \text{ where } \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{\mathcal{V}} \exp(e_{ik})}, \quad e_{ij} = a(s_{i-1}, h_j) \]

- context vector
- node representation
- attention weights
- alignment model
Attention Based Sequence Decoding

\[ c_i = \sum_{j=1}^{\mathcal{V}} \alpha_{ij} h_j, \text{ where } \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{\mathcal{V}} \exp(e_{ik})}, \quad e_{ij} = a(s_{i-1}, h_j) \]

- Context vector
- Node representation
- Attention weights
- Alignment model

• Objective Function

\[ \theta^* = \arg\max_{\theta} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \log p(y_t^n | y_{<t}^n, x^n) \]
Text Reasoning and Shortest Path

garden (A) bathroom (B) bedroom (C)
hallway (D) office (E) kitchen (F)

1. The garden is west of the bathroom.
2. The bedroom is north of the hallway.
3. The office is south of the hallway.
4. The bathroom is north of the bedroom.
5. The kitchen is east of the bedroom.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>west</td>
<td>B</td>
<td>north</td>
<td>D</td>
<td>south</td>
</tr>
</tbody>
</table>

Q: How do you go from the bathroom to the hallway

Q: path(B, D)

<table>
<thead>
<tr>
<th>Model</th>
<th>bAbI T19</th>
<th>SP-S</th>
<th>SP-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>25.2%</td>
<td>8.1%</td>
<td>2.2%</td>
</tr>
<tr>
<td>GGS-NN</td>
<td>98.1%</td>
<td>100.0%</td>
<td>95.2%</td>
</tr>
<tr>
<td>GCN</td>
<td>97.4%</td>
<td>100.0%</td>
<td>96.5%</td>
</tr>
<tr>
<td>Graph2Seq</td>
<td>99.9%</td>
<td>100.0%</td>
<td>99.3%</td>
</tr>
</tbody>
</table>
Effect of Bidirectional Node Embedding

Bidirectional Node Embedding VS Unidirectional Node Embedding

Converge More quickly
When Shall We Use Graph2Seq?

• Case I: the inputs are naturally or best represented in graph

```
describe-01
  :ARG0
  :ARG1
  :ARG2
person
  :name
  op1
  "Ryan"
name
  genius
```

“Ryan’s description of himself: a genius.”

• Case II: Hybrid Graph with sequence and its hidden structural information

```
are
expl
ada
jobs
outside
austin

are there ada jobs outside Austin” with its dependency parsing tree results

compound
case
nsubj
nmod
```
Learning Structured Input-Output Translation

- To bridge the semantic gap between the human-readable words and machine-understandable logics.
- Semantic parsing is important for question answering, text understanding.
- Automatically solving of MWP is a growing interest.

<table>
<thead>
<tr>
<th></th>
<th>Text Input:</th>
<th>Structured output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>what jobs are there for web developer who know ’c++’ ?</td>
<td>answer( A , ( job ( A ) , title ( A , W ) , const ( W , ’Web Developer’ ) ) , language ( A , C ) , const ( C , ’c++’ ) ) )</td>
</tr>
<tr>
<td>MWP</td>
<td>0.5 of the cows are grazing grass . 0.25 of the cows are sleeping and 9 cows are drinking water from the pond . find the total number of cows .</td>
<td>(( ( 0.5 * x ) + ( 0.25 * x ) ) ) + 9.0 = x</td>
</tr>
</tbody>
</table>
Graph and Tree Constructions

Figure 1: Dependency tree augmented text graph

Figure 2: Constituency tree augmented text graph

Figure 3: A sample tree output in our decoding process from expression "( (0.5 * x) + (0.25 * x) ) + 9.0 = x"
Tree Decoding

DFS-based tree decoder

BFS-based tree decoder
Graph-to-Tree Model

Input: are there ada jobs outside austin

Parse

Dependency:
{u'dep': 'compound', u'jobs', u'ada'},
{u'dep': 'nsubj', u'are', u'jobs'},
{u'dep': 'case', u'outside'},

Construct Graph

Graph Encoder

Separated Attention Based Tree Decoding

\[ c_{v_1} = \sum \alpha_{t(v)} z_v, \forall v \in V_1 \]
\[ c_{v_2} = \sum \beta_{t(v)} z_v, \forall v \in V_2 \]

\[ \alpha_{t(v)} = \frac{\exp(score(z_v, s_t))}{\exp(\sum_{k=1}^{V_1} score(z_k, s_t))}, \forall v \in V_1 \]
\[ \beta_{t(v)} = \frac{\exp(score(z_v, s_t))}{\exp(\sum_{k=1}^{V_2} score(z_k, s_t))}, \forall v \in V_2 \]

\[ \tilde{s}_t = \tanh(W_c \cdot [c_{v_1}; c_{v_2}; s_t] + b_c) \]
# Math Word Problem

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAWPS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retrieval</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jaccard</td>
<td>45.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cosine</td>
<td>38.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiLSTM</td>
<td>62.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-attention</td>
<td>60.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2seq</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM</td>
<td>25.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN</td>
<td>44.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Tree</td>
<td>65.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graph2Seq</td>
<td>70.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MathDQN</td>
<td>60.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T-RNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full model</td>
<td>66.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W/o equation normalization</td>
<td>63.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W/o self-attention</td>
<td>66.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group-Att</td>
<td>76.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Graph2Tree</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with constituency graph</td>
<td>78.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>with dependency graph</td>
<td>76.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Solution accuracy comparison on MAWPS

<table>
<thead>
<tr>
<th>Methods</th>
<th>MATHQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2Prog</td>
<td>51.9</td>
</tr>
<tr>
<td>Seq2Prog+Cat</td>
<td>54.2</td>
</tr>
<tr>
<td>TP-N2F</td>
<td>55.95</td>
</tr>
<tr>
<td>Seq2seq</td>
<td>58.36</td>
</tr>
<tr>
<td>Seq2Tree</td>
<td>64.15</td>
</tr>
<tr>
<td>Graph2Seq</td>
<td>65.36</td>
</tr>
<tr>
<td><strong>Graph2Tree</strong></td>
<td></td>
</tr>
<tr>
<td>with constituency graph</td>
<td>69.65</td>
</tr>
<tr>
<td>with dependency graph</td>
<td>65.66</td>
</tr>
</tbody>
</table>

Table 6: Solution accuracy comparison on MATHQA
Visualization of Separated Attentions

(a) A graph-to-tree translation example

(b) Attention for word nodes

(c) Attention for structure nodes

Figure 5: Effect visualization of our separated attentions on both word and structure nodes in a graph.
Half-hour Break

Want to prepare for our demo session?
1) git clone https://github.com/graph4ai/graph4nlp_demo
2) follow Get Started instructions in README

References:
• Graph4NLP demo link: https://github.com/graph4ai/graph4nlp_demo
• Graph4NLP library link: https://github.com/graph4ai/graph4nlp
• DLG4NLP literature link: https://github.com/graph4ai/graph4nlp_literature
DLG4NLP
Applications
Information Extraction
Outline

Semantic Graph Parsing for Event Extraction

• Cross-lingual Structure Transfer for Relation Extraction and Event Extraction

• Cross-media Structured Common Space for Multimedia Event Extraction

• Graph Schema-guided Event Extraction and Prediction

• Cross-media Knowledge Graph based Misinformation Detection
Information Extraction: a Sequence-to-Graph Task

OneIE [Lin et al., ACL2020] framework extracts the information graph from a given sentence in four steps: encoding, identification, classification, and decoding.
Extending to Graph-to-Graph Task

- [Nyuyen et al., NAACL2021]
Moving from Seq-to-Graph to Graph-to-Graph

• [Zhang and Ji, NAACL2021]

• Abstract Meaning Representation (AMR):
  • A kind of rich semantic parsing
  • Converts input sentence into a directed and acyclic graph structure with fine-grained node and edge type labels

• AMR parsing shares inherent similarities with information network (IE output)
  • Similar node and edge semantics
  • Similar graph topology

• Semantic graphs can better capture non-local context in a sentence

• Exploit the similarity between AMR and IE to help on joint information extraction
AMR-IE: An AMR-guided encoding and decoding framework for IE

Input Sentence → Roberta → Encoded Sentence

Pretrained AMR Parser

CRF-based Span Tagger

“create-01”

“Japan”

“insurer-01”

“Business:Start-Org”

“Agent”

“Place”

“Org”

“create”

“AIG”

“Japan”

“insurer”

AMR-Guided Graph Encoding and Decoding

IE Output
AMR Guided Graph Encoding: Using an Edge-Conditioned GAT

- Map each candidate entity and event to AMR nodes.
- Update entity and event representations using an edge-conditioned GAT to incorporate information from AMR neighbors.

\[
\alpha^l_{i,j} = \frac{\exp \left( \sigma \left( f^l[W h^l_i : W_e e_{i,j} : W h^l_j] \right) \right)}{\sum_{k \in \mathcal{N}_i} \exp \left( \sigma \left( f^l[W h^l_i : W_e e_{i,k} : W h^l_k] \right) \right)}
\]

\[
h^* = \sum_{i \in \mathcal{N}_i} \alpha^l_{i,j} h^l_i
\]

\[
h^{l+1} = h^l + \gamma \cdot W^* h^*
\]
AMR Guided Graph Decoding: Ordered decoding guided by AMR

- Beam search based decoding as in OneIE (Lin et al. 2020).
- The decoding order of candidate nodes are determined by the hierarchy in AMR in a **top-to-down manner**.
- For example, the correct ordered decoding in the following graph is:

$$\tau_1, \tau_2, \varepsilon_{1,1}, \varepsilon_{2,1}, \varepsilon_{1,2}, \varepsilon_{2,2}, \varepsilon_{2,3}$$
Examples on how AMR graphs help

<table>
<thead>
<tr>
<th>Sentence</th>
<th>AMR Parsing</th>
<th>OneIE outputs</th>
<th>AMR-IE outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the resolution is not passed, Washington would likely want to use the airspace for strikes against Iraq and for airlifting troops to northern Iraq.</td>
<td>airlift-01</td>
<td>Movement:Transport “airlifting”</td>
<td>Movement:Transport “airlifting”</td>
</tr>
<tr>
<td></td>
<td>“Washington”</td>
<td>“Iraq”</td>
<td>“Washington”</td>
</tr>
<tr>
<td></td>
<td>troop</td>
<td></td>
<td>“troop”</td>
</tr>
<tr>
<td></td>
<td>north</td>
<td></td>
<td>Artifact</td>
</tr>
<tr>
<td>A Pakistani court in central Punjab province has sentenced a Christian man to life imprisonment for a blasphemy conviction, police said Sunday.</td>
<td>cause-01</td>
<td>Justice: Sentence “sentenced”</td>
<td>Justice: Convict “conviction”</td>
</tr>
<tr>
<td></td>
<td>sentence-01</td>
<td>Justice: Convict “conviction”</td>
<td>Justice: Convict “conviction”</td>
</tr>
<tr>
<td></td>
<td>“province”</td>
<td></td>
<td>“court”</td>
</tr>
<tr>
<td></td>
<td>man</td>
<td></td>
<td>“man”</td>
</tr>
<tr>
<td></td>
<td>blasphemy</td>
<td></td>
<td>Artifact</td>
</tr>
<tr>
<td>Russian President Vladimir Putin’s summit with the leaders of Germany and France may have been a failure that proves there can be no long-term “peace camp” alliance following the end of war in Iraq.</td>
<td>fall-01</td>
<td>Contact: Meet “summit”</td>
<td>Contact: Meet “summit”</td>
</tr>
<tr>
<td></td>
<td>“Vladimir Putin”</td>
<td>Entity</td>
<td>Entity</td>
</tr>
<tr>
<td></td>
<td>“prove-01”</td>
<td>“Vladimir Putin”</td>
<td>“leaders”</td>
</tr>
<tr>
<td></td>
<td>“Germany”</td>
<td></td>
<td>“leaders”</td>
</tr>
<tr>
<td></td>
<td>“France”</td>
<td></td>
<td>“leaders”</td>
</tr>
<tr>
<td></td>
<td>“Iraq”</td>
<td></td>
<td>“leaders”</td>
</tr>
<tr>
<td></td>
<td>“Vladimir Putin”</td>
<td></td>
<td>“leaders”</td>
</tr>
<tr>
<td>Major US insurance group AIG is in the final stage of talks to take over General Electric’s Japanese life insurance arm in a deal to create Japan’s sixth largest life insurer, reports said Wednesday.</td>
<td>create-01</td>
<td>Business:Start-Org “create”</td>
<td>Business:Start-Org “create”</td>
</tr>
<tr>
<td></td>
<td>“AIG”</td>
<td>Agent</td>
<td>Agent</td>
</tr>
<tr>
<td></td>
<td>“person”</td>
<td>“AIG”</td>
<td>“insurer”</td>
</tr>
<tr>
<td></td>
<td>“ARG1-of”</td>
<td></td>
<td>“Japan”</td>
</tr>
<tr>
<td></td>
<td>“Japan”</td>
<td></td>
<td>“Japan”</td>
</tr>
</tbody>
</table>
Outline

- Semantic Graph Parsing for Event Extraction
  Cross-lingual structure transfer for Relation Extraction and Event Extraction
- Cross-media Structured Common Space for Multimedia Event Extraction
- Graph Schema-guided Event Extraction and Prediction
- Cross-media Knowledge Graph based Misinformation Detection
Cross-lingual Structure Transfer

The *detainees* were taken to a *processing center*

(teams of doctors were seen in packed emergency rooms)
Graph Convolutional Networks (GCN) Encoder

- Extend the monolingual design (Zhang et al., 2018) to cross-lingual
- Convert a sentence with N tokens into N*N adjacency matrix $A$
- Node: token, each edge is a directed dependency edge
- Initialization of each node’s representation

$$h_i^{(0)} = x_i^w \oplus x_i^p \oplus x_i^d \oplus x_i^e$$

- Word embedding
- POS tag
- Dependency relation
- Entity type

- At the $k^{th}$ layer, derive the hidden representation of each node from the representations of its neighbors at previous layer

$$h_i^{(k)} = \text{ReLU} \left( \sum_{j=0}^{N} \frac{A_{ij} W^{(k)} h_j^{(k-1)}}{d_i + b^{(k)}} \right)$$
Application on Event Argument Extraction

• Task: Classify each pair of event trigger and entity mentions into one of pre-defined event argument roles or NONE

• Max-pooling over the final node representations to obtain representations for sentence, trigger and argument candidate, and concatenate them

• A softmax output layer for argument role labeling

\[ L^a = \sum_{i=1}^{N} \sum_{j=1}^{L_i} y_{ij} \log(\sigma(U^a \cdot [h^t_i; h^s_{ij}; h^a_j])) \]
The detained were taken to a processing center.

Команды врачей были замечены в упакованных отделениях скорой помощи.

(teams of doctors were seen in packed emergency rooms)
Cross-lingual Edge Transfer Performance

- Chinese Event Argument Extraction

![Graph showing F1 score vs. number of event mentions for different languages]

- Trained from English
- Trained from Arabic
- Trained from Chinese
Outline

• Semantic Graph Parsing for Event Extraction
• Cross-lingual structure transfer for Relation Extraction and Event Extraction
  Cross-media Structured Common Space for Multimedia Event Extraction
• Graph Schema-guided Event Extraction and Prediction
• Cross-media Knowledge Graph based Misinformation Detection
Last week, U.S. Secretary of State Rex Tillerson visited Ankara, the first senior administration official to visit Turkey, to try to seal a deal about the battle for Raqqa and to overcome President Recep Tayyip Erdogan's strong objections to Washington's backing of the Kurdish Democratic Union Party (PYD) militias. Turkish forces have attacked SDF forces in the past around Manbij, west of Raqqa, forcing the United States to deploy dozens of soldiers on the outskirts of the town in a mission to prevent a repeat of clashes, which risk derailing an assault on Raqqa.

Output: Multimedia Events & Argument Roles

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Movement.Transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Text Trigger</td>
</tr>
<tr>
<td>Image</td>
<td></td>
</tr>
</tbody>
</table>
Weakly Aligned Structured Embedding

-- Training Phase (Common Space Construction)

Caption
Weakly Aligned Structured Embedding

-- Training and Test Phase (Cross-media shared classifiers)
Compare to Single Data Modality Extraction

- Surrounding sentence helps visual event extraction.
- Image helps textual event extraction.
Compare to Cross-media Flat Representation

<table>
<thead>
<tr>
<th>Model</th>
<th>Event Type</th>
<th>Argument Role</th>
<th>Model</th>
<th>Event Type</th>
<th>Argument Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
<td>Justice.ArrestJail</td>
<td>Agent = man</td>
<td>Flat</td>
<td>Movement.Transport</td>
<td>Artifact = none</td>
</tr>
<tr>
<td>Ours</td>
<td>Justice.ArrestJail</td>
<td>Entity = man</td>
<td>Ours</td>
<td>Movement.Transport</td>
<td>Artifact = man</td>
</tr>
</tbody>
</table>
Outline

• Semantic Graph Parsing for Event Extraction
• Cross-lingual structure transfer for Relation Extraction and Event Extraction
• Cross-media Structured Common Space for Multimedia Event Extraction
  Graph Schema-guided Event Extraction and Prediction
• Cross-media Knowledge Graph based Misinformation Detection
Move from Entity-Centric to Event-Centric NLU
Event Graph Schema Induction

• [Li et al., EMNLP2020]

• How to capture complex connections among events?
  • Temporal relations exist between almost all events, even those that are not semantically related
  • Causal relations have been hobbled by low inter-annotator agreement (Hong et al., 2016)

• Two events are connected through entities and their relations
Event Graph Schema Induction

• History repeats itself: Instance graphs (a) and (b) refer to very different event instances, but they both illustrate a same scenario.

• We select salient and coherent paths based on Path Language Model, and merge them into graph schemas.
Path Language Model

- Path Language Model is trained on two tasks
  - Autoregressive Language Model Loss: capturing the frequency and coherence of a single path
  - Neighbor Path Classification Loss: capturing co-occurrence of two paths
Schema-Guided Information Extraction

- Use the state-of-the-art IE system OneIE (Lin et al., 2020) to decode converts each input document into an IE graph
- Each path in the graph schema is encoded as a single global feature for scoring candidate IE graphs
- OneIE promotes candidate IE graphs containing paths matching schema graphs
- http://blender.cs.illinois.edu/software/oneie
- F-scores (%) on ACE2005 data [Lin et al., ACL2020]:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Entity</th>
<th>Event Trigger Identification</th>
<th>Event Trigger Classification</th>
<th>Event Argument Identification</th>
<th>Event Argument Classification</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>90.3</td>
<td>75.8</td>
<td>72.7</td>
<td>57.8</td>
<td>55.5</td>
<td>44.7</td>
</tr>
<tr>
<td>+PathLM</td>
<td>90.2</td>
<td>76.0</td>
<td>73.4</td>
<td>59.0</td>
<td>56.6</td>
<td>60.9</td>
</tr>
</tbody>
</table>
Temporal Complex Event Schema Composition

- **Graph Structure Aware:**
  - Encode entity coreference and entity relation
  - Capture the interdependency of events and entities (sequences can not)

- **Scenario guided:**
  - Train one model based on instance graphs of the same scenario

- **Probabilistic:**
  - Support downstream tasks, such as event prediction
Generative Event Graph Model

- Schemas are the hidden knowledge to control instance graph generation
- Step 1. Event Node Generation
- Step 2. Message Passing
- Step 3. Argument Node Generation
- Step 4. Relation Edge Generation
- Step 5. Temporal Edge Generation
Schema-guided Event Prediction

- **Schema-guided Event Prediction**: The task aims to predict ending events of each graph.
  - Considering that there can be multiple ending events in one instance graph, we rank event type prediction scores and adopt MRR and HITS@1 as evaluation metrics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Models</th>
<th>MRR</th>
<th>HITS@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Human Schema</td>
<td>0.173</td>
<td>0.205</td>
</tr>
<tr>
<td></td>
<td>Event Graph Model</td>
<td>0.457</td>
<td>0.591</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Models</th>
<th>MRR</th>
<th>HITS@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IED</td>
<td>Human Schema</td>
<td>0.072</td>
<td>0.222</td>
</tr>
<tr>
<td></td>
<td>Event Graph Model</td>
<td>0.203</td>
<td>0.426</td>
</tr>
</tbody>
</table>
Outline

• Semantic Graph Parsing for Event Extraction
• Cross-lingual structure transfer for Relation Extraction and Event Extraction
• Cross-media Structured Common Space for Multimedia Event Extraction
• Graph Schema-guided Event Extraction and Prediction

Cross-media Knowledge Graph based Misinformation Detection
Information Pollution

- Why would anyone ever believe these rumors?
- Because humans are very good at connecting dots
- And perhaps too good →

Graph4NLP
Quiz Time! Which one is Fake News?

Burma’s once-outlawed National League for Democracy is holding its first party congress since the opposition group was founded 25 years ago. Delegates in Rangoon will draw up a policy framework and elect a central committee during the three-day meeting that began Friday. Democracy icon Aung San Suu Kyi is also expected to be reappointed as head of the party. The Nobel laureate helped the NLD to a strong showing in historic April by-elections, which saw the party win 43 of the 45 contested seats. But the NLD is setting its sights on 2015, when it hopes to take power during national elections. But the party faces several challenges as it attempts to fashion itself into a viable political alternative to the military, which still dominates parliament and other government institutions. One of the most pressing issues is electing younger leaders to replace the party’s elderly founding members, many of whom are in their 80s or 90s and in poor health.

Congress delegates prepare to pose for photographs as they arrive to attend the National League for Democracy party’s (NLD) congress in Rangoon, March 8, 2013.

Delegates from the NLD gather in Rangoon for the party’s annual congress. The NLD is headed by Nobel Peace Prize winner Aung San Suu Kyi. The party is expected to win a majority of seats in the parliament.

This year’s NLD Congress is the first time the party has been able to elect its own leadership. Nyan Win, a member of NLD’s executive committee, told VOA that the party is looking forward to the new generation of leaders.

The party has come a long way since the military seized power in 1962. The NLD was founded by a Briton. Since then, Burma has been ruled by a quasi-civilian government. However, the military has still maintained tight control over the country’s political institutions. Phil Robertson, Asia director for Human Rights Watch, said he hopes the party will push forward with reforms that will allow the army to step down and allow the civilian government to take over.
Quiz Time! Which Caption is Fake?

On 24 May 2017 the Philippines militants left their barracks in the outskirts of southern Marawi city to reinforce fellow troops who had been under siege by Islamic troops.

Philippine troops arrive at their barracks to reinforce fellow troops following the siege by Muslim militants, on the outskirts of Marawi city in the southern Philippines, May 24, 2017.
Anis Amri (L), the Tunisian suspect of the Berlin Christmas market attack, is seen in this photo taken from security cameras at the Milan Central Train Station in downtown Milan, Italy December 23, 2016.

Anis Amri, a Tunisian suspected of defending the Christmas market in Milan, was seen in this photo given from a security camera at the Central Train Station of downtown Berlin on 23 December 2016.
Knowledge Element-Level Misinformation Detection [Fung et al., ACL2021]

Motivation: misinformative parts of a fake news article lie along the fine-grained details

Current Issues:
- Fake news detection approaches tend to focus on checking facts, semantic inconsistencies, style or bias, lacking a unified framework.
- The document-level detection task lacks precision and explainability.

Ex of Grover-Generated Fake News - News Spoofing

Hong Kong declared Independence from China Yesterday - February 19, 2021

In a historic decision made yesterday, Hong Kong declared its independence from mainland China. The Senate of Hong Kong, the local government's legislative body, passed the inaugural Resolution of Independence after members of all races, sects and ages gathered in the senate chambers...

"As the Chief Executive Council today endorsed the proposal of the Chief Executive Council to confirm the first proposed Resolution of Independence, Hong Kong is determined to complete the path of self-rule," said London-based broadcaster CNN yesterday...

"We look forward to the motion being made by the Legislative Council and to firmly reaffirming our commitment to a prosperous and stable life of our people, while working together with China," Hong Kong's Chief Executive, Carrie Lam, said in a statement, according to AFP.
Compare with Previous Work

- **Motivation**: misinformative parts of a fake news article lie along the fine-grained details
  - Existing approaches lack a *unified framework* in checking facts, semantic inconsistencies, text features and bias

<table>
<thead>
<tr>
<th></th>
<th>Text Features</th>
<th>Structured Knowledge</th>
<th>Source Bias</th>
<th>Multimedia</th>
<th>Knowledge Element Level Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perez-Rosas et al. (2018)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pan et al. (2017)</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baly et al. (2018)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zellers et al. (2019)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tan et al. (2020)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>InfoSurgeon (Ours)</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Comparison with related work on fake news detection
Knowledge Element-Level Misinformation Detection

- Combine local and global features
- Leverage external knowledge to help pinpoint misinformation

Police brutality has risen to a new, extreme level in HK this past weekend. HK police started shooting at protestors on the streets, including the unarmed, peaceful protestors. One notable incidence involved a woman at the Tsim Sha Tsui bus stop being shot in the eye by a policeman hiding behind corners. No warning was issued beforehand, and the woman was permanently blinded. Local activists are avidly calling for international attention on the HK police brutality.
Knowledge Element-Level Misinformation Detection

We also propose a new task in addition to document-level fake news detection that is more challenging but interesting.

Label each triplet connecting two entities as True/False.
Multimedia Knowledge Graph Construction

Image Caption
Hong Kong police shoot real bullets at protestors from hidden corners.

T = True, F = False, ◼ = entity ▲ = event trigger
Graph Propagation

Yi Fung, Christopher Thomas, Revanth Gangi Reddy, Sandeep Polisetty, Heng Ji, Shih-Fu Chang, Kathleen McKeown, Mohit Bansal and Avi Sil

“InfoSurgeon: Cross-Media Fine-grained Information Consistency Checking for Fake News Detection”. In submission to ACL 2021
Annotating specifically which elements in a KG are fake is time-intensive/difficult.

We propose a solution to **automatically obtain knowledge-element labeled knowledge graphs for free**.

Given a set of real news articles, we extract KGs from the real articles.

Train a text generator model that learns to recreate an article from its KG.

To generate fake data, perform manipulation operations on the KG (editing knowledge relations, events, entities, etc.) to produce KG’.

Generate a fake article from KG’.

Key insight – We now know specifically which elements in KG’ were manipulated!
Manipulated KG-to-Article Synthesis

We perform the following manipulations on KGs:

- **Entity swapping** – Swapping entity that has same type and similar embedding (so they are harder to tell apart)

- **Addition of new relation or event** – Randomly select relation / event argument roles and append a new entity to the relation / event

- **Subgraph replacement** – Select a subgraph of the news article from an entity and replace it with a subgraph from another news article
We manipulate knowledge graphs to synthesize fake news which contain known types of inconsistencies.

This example is trivially detectable due to an inconsistency with the image.

The Zambia-based Bamboo truck is the first of its kind in the world, and it’s designed to be a cheaper and lighter alternative to the traditional, heavy-duty, imported, metal ambulance. The Zambian-designed truck is built of bamboo, which is an environmentally friendly material.
• By imposing cross-media knowledge graph manipulation constraints, we prevent generating text with obvious inconsistencies.

• Enables generating more realistic / challenging data for training detector
Caption Manipulation – AMR-to-Caption Synthesis

• Use text parser to get AMR graphs (Banarescu et al., 2013) from captions
• Use AMR since they capture fine-grained relations expressing who does what to whom
• Manipulations:
  • **Role switching** – Swapping entity positions in AMR graph
  • **Predicate negation** – Replace triggers / verbs with antonyms from WordNet
• Use off-the-shelf model for AMR to text synthesis (Ribeiro et al, 2020)

**True Caption:**
In Afghanistan, the Taliban released to the **media** this picture, which it said shows the suicide bombers who **attacked** the **army** base in Mazar-i-Sharif, April 21, 2017

**Fake caption:**
On 21 April 2017 the Taliban released this picture to the **army** in Afghanistan which they said was a suicide bomber **hiding** at a **media** base in the city of Mazar-i-Sharif

• **Ethical Statement:** we are not going to share our generator, but sharing our detector!
Knowledge Element-Level Misinformation Dataset

• To address the lack of data for the detection task, we further contribute a KG2txt fake news generation approach, which allows for control over knowledge element manipulation and creating silver standard annotation data.

<table>
<thead>
<tr>
<th></th>
<th>Overall</th>
<th>Real Documents</th>
<th>Fake Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Detection Accuracy</td>
<td>61.3%</td>
<td>80.4%</td>
<td>42.3%</td>
</tr>
</tbody>
</table>

The Turing Test results above show that our automatically generated fake documents are also very hard for humans to detect.
# Knowledge Element-Level Misinformation Detection

Experimental result on traditional document-level detection:

<table>
<thead>
<tr>
<th></th>
<th>NYTimes Neural News Dataset</th>
<th>VOA Manipulated KG2Txt Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grover</td>
<td>56.0%</td>
<td>86.4%</td>
</tr>
<tr>
<td>DIDAN</td>
<td>77.6%</td>
<td>88.3%</td>
</tr>
<tr>
<td>InfoSurgeon (Our Model)</td>
<td>94.5%</td>
<td>92.1%</td>
</tr>
</tbody>
</table>

Experimental result on the novel task, knowledge element level misinformation detection:

<table>
<thead>
<tr>
<th></th>
<th>VOA Manipulated KG2Txt Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random (baseline)</td>
<td>27%</td>
</tr>
<tr>
<td>InfoSurgeon (Our Model)</td>
<td>37%</td>
</tr>
</tbody>
</table>
### A Successful Example

- Example of fake news article in which baseline misses, but *InfoSurgeon* successfully detects

<table>
<thead>
<tr>
<th>Image</th>
<th>Caption</th>
<th>Body Text</th>
<th>Misinformative KEs</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Fort McHenry" /></td>
<td>Aerial view of <em>Fort McHenry</em>.</td>
<td>The battle of <em>Fort McHenry</em>, which took place in September of 1814, was a pivotal moment in the U.S. War of Independence...When the <em>British</em> finally left, they left behind a trail of destruction, including the destruction of the <em>twin towers</em> of the World Trade Center...</td>
<td><code>&lt;British, Conflict.Attack, twin_towers&gt;</code></td>
</tr>
</tbody>
</table>
Demo 1: Multimedia Event Recommendation

(Li et al., ACL2020 Best Demo Paper Award)
GitHub: https://github.com/GAIA-IE/gaia
DockerHub: https://hub.docker.com/orgs/blenderl-nlp/repositories
Demo: http://159.89.180.81/demo/video_recommendation/index_attack_dark.html
Demo 2: Event Heatmap for Disaster Relief

- Re-trainable Systems: http://159.89.180.81:3300/elisa_ie/api
- Demos: http://159.89.180.81:3300/elisa_ie
- Heat map: http://159.89.180.81:8080/
Software and Resources

- KAIROS RESIN Cross-document Cross-lingual Cross-media Information Extraction system (Wen et al., NAACL2021 demo)
  - https://github.com/RESIN-KAIROS/RESIN-pipeline-public
- Joint Neural Information Extraction system (Lin et al., ACL2020)
  - http://blender.cs.illinois.edu/software/oneie/
- GAIA Multimedia Event Extraction system and new benchmark with annotated data set (Li et al., ACL2020 demo)
  - GitHub: https://github.com/GAIA-AIDA/uiuc_ie_pipeline_fine_grained
  - Text IE DockerHub: https://hub.docker.com/orgs/blenderlplp/
  - Visual IE repositories: https://hub.docker.com/u/dannapierskitoptal
Text Clustering and Matching
Information Explosion
Disadvantages of existing systems

- Messed document lists
- Extremely fine-grained (articles)
- Redundant useless information
- Unstructured information
Detect events automatically from massive news articles

Story Forest

Trees denotes stories, nodes denotes events

Edges in the tree denotes events evolving relationship

Liu et al., Story Forest: Extracting Events and Telling Stories from Breaking News, TKDD’20
Story Forest System

Prepare Data

Preprocessing Documents
1. Document filtering
2. Word segmentation

Keyword Extraction
1. Extract a variety of word features
2. Classify whether each word is a keyword

Keyword Graph

Construct Keyword Graph
Construct or update keyword graph by keyword co-occurrence in new incoming documents.

Split Keyword Graph
1. Identify changed part of keyword graph
2. Community detection
3. Filtering out small sub-graphs

Grow Stories

Grow Story Forest
1. Compare new events with existing story nodes
2. Merge same events, or insert events to stories

Find Related Story
1. Identify candidate stories
2. Find most related story
3. If no related story, create a new story

Cluster Events

Second Layer Clustering
1. Doc-pair relationship classification
2. Construct doc graph
3. Community detection on doc graph

First Layer Clustering
Cluster new documents by keyword communities.

Keywords

Documents

Events

Stories

Liu et al., Story Forest: Extracting Events and Telling Stories from Breaking News, TKDD’20
Applied to QQ browser
hot topic list

Hawking public PhD thesis
Text Matching Tasks

Ad-hoc Retrieval
- Web Search
- Academic Search
- Email Search
- Twitter / Weibo Search
- Question Answering
- Query Suggestion

Long Text Matching
- Story Forest Formation
- Article Recommendation
- Citation Recommendation
- Genre Classification
- Sentiment Analysis
- Spam Filtering
- Document Classification

Short Source
- Google
- Bing
- Baidu

Long Target
Why Long Document Matching

Identify the relationship between documents

Same event? Related events?
**Divide-and-Conquer**

**Our strategies**

- **Divide**
  - 1
  - 2

- **Align**
  - 1
  - 2

- **Distributed match**
  - 1
  - 2

**Limitations**

- **Hard to encode**
- **Flexible order**
- **Time complexity**
Rick asks Morty to travel with him in the universe. Morty doesn’t want to go as Rick always brings him dangerous experiences. However, the destination of this journey is the Candy Planet, which is a fascinating place that attracts Morty. The planet is full of delicious candies. Summer wishes to travel with Rick. However, Rick doesn’t like to travel with Summer.
Graph Decomposition for Document Matching

(a) Construct CIG

(b) Local Matching

(c) Aggregate

(d) Classify/Score

\[ \text{Doc A} \quad \text{Doc B} \]

\[ \text{Concept Interaction Graph} \]

\[ \text{Sentences 1, Sentence 2} \]

\[ \text{Feature Extractor} \]

\[ \text{Sentence 1, Sentence 2} \]

\[ \text{GCN Layers} \]

\[ \text{Local matching} \]

\[ \text{Global matching} \]

\[ \text{Classify} \]

\[ \text{Global matching} \]

\[ \text{Result} \]

Liu et al., Matching article pairs with graphical decomposition and convolutions, ACL’19
### Experiments

<table>
<thead>
<tr>
<th>Baselines</th>
<th>CNSE Acc</th>
<th>CNSE F1</th>
<th>CNSS Acc</th>
<th>CNSS F1</th>
<th>Our models</th>
<th>CNSE Acc</th>
<th>CNSE F1</th>
<th>CNSS Acc</th>
<th>CNSS F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. ARC-I</td>
<td>53.84</td>
<td>48.68</td>
<td>50.10</td>
<td>66.58</td>
<td>XI. CIG-Siam</td>
<td>74.47</td>
<td>73.03</td>
<td>75.32</td>
<td>78.58</td>
</tr>
<tr>
<td>II. ARC-II</td>
<td>54.37</td>
<td>36.77</td>
<td>52.00</td>
<td>53.83</td>
<td>XII. CIG-Siam-GCN</td>
<td>74.58</td>
<td>73.69</td>
<td>78.91</td>
<td>80.72</td>
</tr>
<tr>
<td>III. DUET</td>
<td>55.63</td>
<td>51.94</td>
<td>52.33</td>
<td>60.67</td>
<td>XIII. CIG$_{cd}$-Siam-GCN</td>
<td>73.25</td>
<td>73.10</td>
<td>76.23</td>
<td>76.94</td>
</tr>
<tr>
<td>IV. DSSM</td>
<td>58.08</td>
<td>64.68</td>
<td>61.09</td>
<td>70.58</td>
<td>XIV. CIG-Sim</td>
<td>72.58</td>
<td>71.91</td>
<td>75.16</td>
<td>77.27</td>
</tr>
<tr>
<td>V. C-DSSM</td>
<td>60.17</td>
<td>48.57</td>
<td>52.96</td>
<td>56.75</td>
<td>XV. CIG-Sim-GCN</td>
<td>83.35</td>
<td>80.96</td>
<td>87.12</td>
<td>87.57</td>
</tr>
<tr>
<td>VI. MatchPyramid</td>
<td>66.36</td>
<td>54.01</td>
<td>62.52</td>
<td>64.56</td>
<td>XVI. CIG$_{cd}$-Sim-GCN</td>
<td>81.33</td>
<td>78.88</td>
<td>86.67</td>
<td>87.00</td>
</tr>
<tr>
<td>VII. BM25</td>
<td>69.63</td>
<td>66.60</td>
<td>67.77</td>
<td>70.40</td>
<td>XVII. CIG-Sim$_{cd}$-GCN</td>
<td>84.64</td>
<td>82.75</td>
<td>89.77</td>
<td>90.07</td>
</tr>
<tr>
<td>VIII. LDA</td>
<td>63.81</td>
<td>62.44</td>
<td>62.98</td>
<td>69.11</td>
<td>XVIII. CIG-Sim$_{cd}$-GCN-Sim$^g$</td>
<td>84.21</td>
<td>82.46</td>
<td>90.03</td>
<td>90.29</td>
</tr>
<tr>
<td>IX. SimNet</td>
<td>71.05</td>
<td>69.26</td>
<td>70.78</td>
<td>74.50</td>
<td>XIX. CIG-Sim$_{cd}$-GCN-BERT$^g$</td>
<td>84.68</td>
<td>82.60</td>
<td>89.56</td>
<td>89.97</td>
</tr>
<tr>
<td>X. BERT fine-tuning</td>
<td>81.30</td>
<td>79.20</td>
<td>86.64</td>
<td>87.08</td>
<td>XX. CIG-Sim$_{cd}$-GCN-Sim$^g$&amp;BERT$^g$</td>
<td>84.61</td>
<td>82.59</td>
<td>89.47</td>
<td>89.71</td>
</tr>
</tbody>
</table>

**Graph Representation:** greatly improves performance. (IX vs. XI) (+4% Acc, F1)

**Graph Convolution:** greatly improves performance. (XIV vs. XV) (+10% Acc, F1)

Liu et al., *Matching article pairs with graphical decomposition and convolutions*, ACL’19
Text Mining and Classification
What are users interested in?

Query: “Theresa May’s resignation speech”

Infer users’ interests
What are users interested in?

Inaccurate recommendation

I don’t care

Articles about Theresa May

Query: “Theresa May’s resignation speech”
What are users interested in?

Monotonous recommendation

Articles about Theresa May’s resignation speech

Query: “Theresa May’s resignation speech”
What are users interested in?

**Good recommendation**

Articles about Brexit Negotiation

Query: “Theresa May’s resignation speech”
What we need?

User interests in a suitable granularity

Relationships between user interests

Brexit Negotiation

Theresa May’s resignation speech
What do people care about: Events

**Event:** a real-world incident that involves specific persons, organizations, or entities, with a certain time/location of occurrence

Theresa May’s resignation speech  

Apple iPhone7 launch
What do people care about: Concepts

- Fuel-efficient cars
- Economy cars
- Marvel heroes
- Revengers

**Concept:** a collection of things that share some common attributes
Attention ontology

Create a web-scale ontology to represent user interests and document topics.

Liu et al., GIANT: Scalable Creation of a Web-scale Ontology, SIGMOD’20
GIANT system

Search

Recommend

Tag

Search Click Graph

Query-Doc Clusters

User Attention Ontology

Document Tagging

Story Composition

Query Conceptualization

Action

Attention

Application

Liu et al., GIANT: Scalable Creation of a Web-scale Ontology, SIGMOD’20
Heterogeneous phrase mining

Query: What are the Hayao Miyazaki’s animated film (有哪些 宫崎骏 的 动画 电影)
Titles: Review Hayao Miyazaki’s animated film (盘点 宫崎骏 动画 电影)
The famous animated films of Hayao Miyazaki (宫崎骏 著名的 动画 电影)
What are the classic Miyazaki’s movies? (有哪些 经典的 宫崎骏 的 电影？)
Concept: Hayat Miyazaki animated film (宫崎骏 动画 电影)

Characteristics of output words
Patterns
Show up multiple times
NER/Part-of-Speech tags
Continuous chunk
Syntactic dependency

Liu et al., GIANT: Scalable Creation of a Web-scale Ontology, SIGMOD’20
Heterogeneous phrase mining

Patterns: graph structure
Show up multiple times: node feature
NER/Part-of-Speech tags: node feature
Continuous chunk: : seq edge
Syntactic dependency: syntactic edge

Liu et al., GIANT: Scalable Creation of a Web-scale Ontology, SIGMOD’20
Heterogeneous phrase mining

Classify Node: Relational GCN
Sort Node: Asymmetric Traveling Salesman Problem

Liu et al., GIANT: Scalable Creation of a Web-scale Ontology, SIGMOD’20
Transductive Text Classification with TextGCN

- **Nodes**: words and documents
- **Edges**: co-occurrence (word-word) and TFIDF (word-doc)
- Model the graph with a **Graph Convolutional Network** (GCN) (Kipf and Welling 2017)

\[ A_{ij} = \begin{cases} 
\text{PMI}(i, j) & i, j \text{ are words, } \text{PMI}(i, j) > 0 \\
\text{TF-IDF}_{ij} & i \text{ is document, } j \text{ is word} \\
1 & i = j \\
0 & \text{otherwise}
\end{cases} \]

Yao et al., *Graph Convolutional Networks for Text Classification*, AAAI’19
Transductive Text Classification with TextGCN

Test accuracy by varying training data promotions

The t-SNE visualization of test document embeddings in 20NG
Inductive Text Classification with TextING

- **Nodes**: words in a document
- **Edges**: co-occurrence (word-word)
- Model the graph with a **Gated Graph Neural Networks** (Li et al., 2015)
- Each document is an individual graph and text level word interactions can be learned in it.
- It can generalise to new words that absent in training, therefore applicable for inductive circumstances.

Zhang et al., *Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks*, ACL’20
Inductive Text Classification with TextING

<table>
<thead>
<tr>
<th>Model</th>
<th>MR*</th>
<th>Ohsumed*</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextGCN</td>
<td>53.15</td>
<td>47.24</td>
</tr>
<tr>
<td>TextING</td>
<td>64.43</td>
<td>57.11</td>
</tr>
</tbody>
</table>

# Words in Training: 465, 7,009
# New Words in Test: 18,299, 7,148

Accuracy (%) of TextGCN and TextING on MR and Ohsumed

Test performance and gain with different percent of training data on MR.

Zhang et al., *Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks*, ACL’20
Applied to feeds news recommendation

Title
See these cars with less than 2L/100km fuel consumption and up to 1000km recharge mileage

Tagged Concept
Low fuel consumption cars

Liu et al., **GIANT: Scalable Creation of a Web-scale Ontology**, SIGMOD’20
Summary

GNN enables:
• Encode multi-scale information
• Encode heterogenous information
Natural Language Generation
Machine Translation
Natural Question Generation

• Input
  • A text passage $X^p = \{x^p_1, x^p_2, ..., x^p_N\}$
  • A target answer $X^a = \{x^a_1, x^a_2, ..., x^a_L\}$

• Output
  • A natural language question

$$\hat{Y} = \{y_1, y_2, ..., y_T\}$$

which maximizes the conditional likelihood

$$\hat{Y} = \arg \max_Y P(Y|X^p, X^a)$$
RL-based Graph2Seq for QG [Chen et al. ICLR’20]

RL-based Graph2Seq for QG [Chen et al. ICLR’20]

Two graph construction strategies:
1) Syntax-based static passage graph construction
2) Semantics-aware dynamic passage graph construction
RL-based Graph2Seq for QG [Chen et al. ICLR’20]

Bi-Fuse GGNN as the graph encoder
### RL-based Graph2Seq for QG [Chen et al. ICLR’20]

Ablation study on the SQuAD split-2 test set.

<table>
<thead>
<tr>
<th>Methods</th>
<th>BLEU-4</th>
<th>Methods</th>
<th>BLEU-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2S_{dyn}+BERT+RL</td>
<td>18.06</td>
<td>G2S_{dyn} w/o feat</td>
<td>16.51</td>
</tr>
<tr>
<td>G2S_{sta}+BERT+RL</td>
<td>18.30</td>
<td>G2S_{sta} w/o feat</td>
<td>16.65</td>
</tr>
<tr>
<td>G2S_{sta}+BERT-fixed+RL</td>
<td>18.20</td>
<td>G2S_{dyn} w/o DAN</td>
<td>12.58</td>
</tr>
<tr>
<td>G2S_{dyn}+BERT</td>
<td>17.56</td>
<td>G2S_{sta} w/o DAN</td>
<td>12.62</td>
</tr>
<tr>
<td>G2S_{sta}+BERT</td>
<td>18.02</td>
<td>G2S_{sta} w/ DAN-word only</td>
<td>15.92</td>
</tr>
<tr>
<td>G2S_{sta}+BERT-fixed</td>
<td>17.86</td>
<td>G2S_{sta} w/ DAN-contextual only</td>
<td>16.07</td>
</tr>
<tr>
<td>G2S_{dyn}+RL</td>
<td>17.18</td>
<td>G2S_{sta} w/ GGNN-forward</td>
<td>16.53</td>
</tr>
<tr>
<td>G2S_{sta}+RL</td>
<td>17.49</td>
<td>G2S_{sta} w/ GGNN-backward</td>
<td>16.75</td>
</tr>
<tr>
<td>G2S_{dyn}</td>
<td>16.81</td>
<td>G2S_{sta} w/ BiGGNN, w/ Seq2Seq</td>
<td>16.14</td>
</tr>
<tr>
<td>G2S_{sta}</td>
<td>16.96</td>
<td>G2S_{sta} w/o BiGGNN, w/ GCN</td>
<td>14.47</td>
</tr>
</tbody>
</table>

- **Bidirectional GNN** performs better
- **Graph2Seq** performs better than Seq2Seq
- Static graph construction performs slightly better
Natural Question Generation From KG

Q: What languages are spoken in Norway?

• Input
  • A KG subgraph $\mathcal{G}$ (i.e., a collection of subject-predicate-object triples)
  • A target answer set $V^\alpha$

• Output
  • A natural language question
    $$\hat{Y} = \{y_1, y_2, ..., y_T\}$$
    which maximizes the conditional likelihood
    $$\hat{Y} = \arg\max_Y P(Y|\mathcal{G}, V^\alpha)$$
Graph2Seq for QG from KG [Chen et al. arXiv’20]

Graph2Seq for QG from KG [Chen et al. arXiv’20]

- **Homogeneous graph?**
  - NO
  - Convert to homogeneous graph?
    - YES
    - Single node type?
      - NO
      - Multi-relational GNNs
        - Bi-Fuse GGNN w/ edge embeddings
      - YES
        - Bi-Fuse GGNN
    - YES
      - Single node type?
        - NO
        - Ignore node/edge types, Levi graph, ...
          - YES
          - Undirected graph?
            - NO
            - Bidirectional?
              - NO
              - Bi-Fuse GGNN
            - YES
              - Bi-Fuse GGNN
          - YES
            - Bidirectional?
              - NO
              - Bi-Fuse GGNN
            - YES
              - Bi-Fuse GGNN
- Static Graph Construction
- Graph embeddings
- Bi-Fuse GGNN
- Graph2Seq
- Dependency Graph
- Constituency Graph
- AMR Graph
- IE Graph
- Topic Graph
- SQL Graph
- Syntax
- Semantics
- Graph
- Similarity
- Similarity Graph
- Co-occurrence
- Co-occurrence Graph
- World Knowledge
- Knowledge Graph
- Application driven
- Graph embeddings
- Bi-Fuse GGNN
- Graph
- Logic
- SQL
- IE
- AMR
- Syntax
- Dependency
- Constituency
- World Knowledge
- Co-occurrence
- Co-occurrence Graph
Graph2Seq for QG from KG [Chen et al. arXiv’20]

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU-4</th>
<th>WQ METEOR</th>
<th>ROUGE-L</th>
<th>BLEU-4</th>
<th>PQ METEOR</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2A</td>
<td>6.01</td>
<td>25.24</td>
<td>26.95</td>
<td>17.00</td>
<td>19.72</td>
<td>50.38</td>
</tr>
<tr>
<td>Transformer</td>
<td>8.94</td>
<td>13.79</td>
<td>32.63</td>
<td>56.43</td>
<td>43.45</td>
<td>73.64</td>
</tr>
<tr>
<td>MHQG+AE</td>
<td>11.57</td>
<td>29.69</td>
<td>35.53</td>
<td>25.99</td>
<td>33.16</td>
<td>58.94</td>
</tr>
<tr>
<td>G2S+AE</td>
<td>29.45</td>
<td>30.96</td>
<td><strong>55.45</strong></td>
<td>61.48</td>
<td>44.57</td>
<td><strong>77.72</strong></td>
</tr>
<tr>
<td>G2S_{edge}+AE</td>
<td>29.40</td>
<td><strong>31.12</strong></td>
<td>55.23</td>
<td>59.59</td>
<td><strong>44.70</strong></td>
<td>75.20</td>
</tr>
</tbody>
</table>

Automatic evaluation results on the WQ and PQ test sets.

Levi graph conversion + homogeneous GNN performs comparably with multi-relational GNN.
Graph2Seq for QG from KG [Chen et al. arXiv’20]

Ablation study on directionality on the PQ test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bidirectional</td>
<td>61.48</td>
<td>44.57</td>
<td>77.72</td>
</tr>
<tr>
<td>Forward</td>
<td>59.59</td>
<td>42.72</td>
<td>75.82</td>
</tr>
<tr>
<td>Backward</td>
<td>59.12</td>
<td>42.66</td>
<td>75.03</td>
</tr>
</tbody>
</table>

Bidirectional GNN performs better
Summarization

- Input
  - A document, dialogue, code or multiple ones
- Output
  - A succinct sentence or paragraph

GNN for Code Summarization [Liu et al. ICLR’21]

GNN for Code Summarization [Liu et al. ICLR’21]

Source Code → Retrieval-augmented Graph → Attention-based Graph

Static CPG graph + attention-based dynamic dynamic

Dynamic Graph Construction

Graph Similarity Metric Learning Techniques
- Node Embedding Based Similarity Metric Learning
- Structure-aware Similarity Metric Learning

Graph Sparsification Techniques
- KNN-style Sparsification
- Epsilon-neighborhood Sparsification
- Graph Regularization

Combining Intrinsic Graph Structures and Implicit Graph Structures
- Joint Learning of Graph Structures and Representations
- Adaptive Learning of Graph Structures and Representations
- Iterative Learning of Graph Structures and Representations

Learning Paradigms
- Static Graph Construction
- Syntax
- Semantics
- Application-driven
- Co-occurrence
- Co-occurrence Graph
- World Knowledge
- Knowledge Graph
GNN for Code Summarization [Liu et al. ICLR’21]

Hybrid GNN running message passing on static & dynamic graphs
## GNN for Code Summarization

[Liu et al. ICLR’21]

<table>
<thead>
<tr>
<th>Methods</th>
<th>In-domain</th>
<th></th>
<th>Out-of-domain</th>
<th></th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU-4</td>
<td>ROUGE-L</td>
<td>METEOR</td>
<td>BLEU-4</td>
<td>ROUGE-L</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>15.20</td>
<td>27.98</td>
<td>13.74</td>
<td>5.50</td>
<td>15.37</td>
</tr>
<tr>
<td>NNGen</td>
<td>15.97</td>
<td>28.14</td>
<td>13.82</td>
<td>5.74</td>
<td>16.33</td>
</tr>
<tr>
<td>Hybrid-DRL</td>
<td>9.29</td>
<td>30.00</td>
<td>12.47</td>
<td>6.30</td>
<td>24.19</td>
</tr>
<tr>
<td>Transformer</td>
<td>12.91</td>
<td>28.04</td>
<td>13.83</td>
<td>5.75</td>
<td>18.62</td>
</tr>
<tr>
<td>Rencos</td>
<td>14.80</td>
<td>31.41</td>
<td>14.64</td>
<td>7.54</td>
<td>23.12</td>
</tr>
<tr>
<td>GCN2Seq</td>
<td>9.79</td>
<td>26.59</td>
<td>11.65</td>
<td>4.06</td>
<td>18.96</td>
</tr>
<tr>
<td>GAT2Seq</td>
<td>10.52</td>
<td>26.17</td>
<td>11.88</td>
<td>3.80</td>
<td>16.94</td>
</tr>
<tr>
<td>SeqGNN</td>
<td>10.51</td>
<td>29.84</td>
<td>13.14</td>
<td>4.94</td>
<td>20.80</td>
</tr>
<tr>
<td><strong>HGNN w/o aug</strong> &amp; static</td>
<td>11.75</td>
<td>29.59</td>
<td>13.86</td>
<td>5.57</td>
<td>22.14</td>
</tr>
<tr>
<td><strong>HGNN w/o aug</strong> &amp; dynamic</td>
<td>11.85</td>
<td>29.51</td>
<td>13.54</td>
<td>5.45</td>
<td>21.89</td>
</tr>
<tr>
<td><strong>HGNN w/o aug</strong></td>
<td>12.33</td>
<td>29.99</td>
<td>13.78</td>
<td>5.45</td>
<td>22.07</td>
</tr>
<tr>
<td><strong>HGNN w/o static</strong></td>
<td>15.93</td>
<td>33.67</td>
<td>15.67</td>
<td>7.72</td>
<td>24.69</td>
</tr>
<tr>
<td><strong>HGNN w/o dynamic</strong></td>
<td>15.77</td>
<td>33.84</td>
<td>15.67</td>
<td>7.64</td>
<td>24.72</td>
</tr>
<tr>
<td><strong>HGNN</strong></td>
<td><strong>16.72</strong></td>
<td><strong>34.29</strong></td>
<td><strong>16.25</strong></td>
<td><strong>7.85</strong></td>
<td><strong>24.74</strong></td>
</tr>
</tbody>
</table>

Combining static + dynamic graphs performs better

Automatic evaluation results (in %) on the CCSD test set.
Machine Translation

- **Input**
  - Source language text $X = \{x_1, x_2, ..., x_N\}$

- **Output**
  - Target language text
    $$\hat{Y} = \{y_1, y_2, ..., y_T\}$$
  which maximizes the conditional likelihood
    $$\hat{Y} = \arg\max_Y P(Y|X)$$

Ref: https://ciklopea.com/blog/translation/science-or-fiction-machine-translation-explained/
Syntactic GCN for MT [Bastings et al. EMNLP’17]

Figure 2: A 2-layer syntactic GCN on top of a convolutional encoder. Loop connections are depicted with dashed edges, syntactic ones with solid (dependents to heads) and dotted (heads to dependents) edges. Gates and some labels are omitted for clarity.
Syntactic GCN for MT [Bastings et al. EMNLP’17]

Graph

Homogeneous graph?

NO

Convert to homogeneous graph?

NO

Single node type?

YES

Multi-relational GNNs

Graph embeddings
Syntactic GCN for MT [Bastings et al. EMNLP’17]

<table>
<thead>
<tr>
<th>Method</th>
<th>Kendall</th>
<th>BLEU₁</th>
<th>BLEU₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW</td>
<td>0.3352</td>
<td>40.6</td>
<td>9.5</td>
</tr>
<tr>
<td>+ GCN</td>
<td>0.3520</td>
<td>44.9</td>
<td>12.2</td>
</tr>
<tr>
<td>CNN</td>
<td>0.3601</td>
<td>42.8</td>
<td>12.6</td>
</tr>
<tr>
<td>+ GCN</td>
<td>0.3777</td>
<td>44.7</td>
<td>13.7</td>
</tr>
<tr>
<td>BiRNN</td>
<td>0.3984</td>
<td>45.2</td>
<td>14.9</td>
</tr>
<tr>
<td>+ GCN</td>
<td>0.4089</td>
<td>47.5</td>
<td>16.1</td>
</tr>
<tr>
<td>BiRNN (full)</td>
<td>0.5440</td>
<td>53.0</td>
<td>23.3</td>
</tr>
<tr>
<td>+ GCN</td>
<td>0.5555</td>
<td>54.6</td>
<td>23.9</td>
</tr>
</tbody>
</table>

Test results for English-German.

<table>
<thead>
<tr>
<th>Method</th>
<th>Kendall</th>
<th>BLEU₁</th>
<th>BLEU₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW</td>
<td>0.2498</td>
<td>32.9</td>
<td>6.0</td>
</tr>
<tr>
<td>+ GCN</td>
<td>0.2561</td>
<td>35.4</td>
<td>7.5</td>
</tr>
<tr>
<td>CNN</td>
<td>0.2756</td>
<td>35.1</td>
<td>8.1</td>
</tr>
<tr>
<td>+ GCN</td>
<td>0.2850</td>
<td>36.1</td>
<td>8.7</td>
</tr>
<tr>
<td>BiRNN</td>
<td>0.2961</td>
<td>36.9</td>
<td>8.9</td>
</tr>
<tr>
<td>+ GCN</td>
<td>0.3046</td>
<td>38.8</td>
<td>9.6</td>
</tr>
</tbody>
</table>

Test results for English-Czech.

Syntactic GCN is helpful
Figure. The multi-modal graph for an input sentence-image pair. The blue and green solid circles denote textual nodes and visual nodes respectively. An intra-modal edge (dotted line) connects two nodes in the same modality, and an inter-modal edge (solid line) links two nodes in different modalities. Note that we only display edges connecting the textual node “playing” and other textual ones for simplicity.
Multi-modal Machine Translation [Yin et al. ACL’20]

Graph attention for inter-modal fusion

---

Graph

Homogeneous graph?

Convert to homogeneous graph?

Single node type?

Graph embeddings

Heterogeneous GNNs
## Multi-modal Machine Translation [Yin et al. ACL’20]

<table>
<thead>
<tr>
<th>Model</th>
<th>En$\Rightarrow$Fr</th>
<th>Test2016</th>
<th>Test2017</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BLEU</td>
<td>METEOR</td>
</tr>
<tr>
<td><strong>Existing Multi-modal NMT Systems</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fusion-conv(RNN) (Caglayan et al., 2017)</td>
<td></td>
<td>53.5</td>
<td>70.4</td>
</tr>
<tr>
<td>Trg-mul(RNN)(Caglayan et al., 2017)</td>
<td></td>
<td>54.7</td>
<td>71.3</td>
</tr>
<tr>
<td>Deliberation Network(TF) (Ive et al., 2019)</td>
<td></td>
<td>59.8</td>
<td>74.4</td>
</tr>
<tr>
<td><strong>Our Multi-modal NMT Systems</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformer (Vaswani et al., 2017)</td>
<td></td>
<td>59.5</td>
<td>73.7</td>
</tr>
<tr>
<td>ObjectAsToken(TF) (Huang et al., 2016)</td>
<td></td>
<td>60.0</td>
<td>74.3</td>
</tr>
<tr>
<td>Enc-att(TF) (Delbrouck and Dupont, 2017b)</td>
<td></td>
<td>60.0</td>
<td>74.3</td>
</tr>
<tr>
<td>Doubly-att(TF) (Helcl et al., 2018)</td>
<td></td>
<td>59.9</td>
<td>74.1</td>
</tr>
<tr>
<td><strong>Our model</strong></td>
<td></td>
<td><strong>60.9</strong></td>
<td><strong>74.9</strong></td>
</tr>
</tbody>
</table>
Hands-on Demonstration
Graph4NLP: A Library for Deep Learning on Graphs for NLP
Overall Architecture of Graph4NLP Library

Dive Into Graph4NLP Library
Data Flow of Graph4NLP

1. **Raw Data**
   - Graph Construction
   - Featured Structured Data (Graph4NLP.GraphData)

2. **Encoded Structured Data** (Graph4NLP.GraphData)
   - GNN Embedding Methods

3. **Prediction**
   - User Model
   - Evaluation
   - Loss

4. **Results**
Computing Flow of Graph4NLP

1. **Text Sequence** → **Static Graph Construction**
2. **Static Graph Construction** → **Graph Data**
   - i. Topology construction
   - ii. Embedding construction
3. **Graph Data** → **Dynamic Graph Construction**
4. **Dynamic Graph Construction** → **Node/Graph Embedding**
   - With mechanisms like attention, copy, coverage, etc.
5. **Node/Graph Embedding** → **Node/Link/Graph Prediction**
6. **Node/Link/Graph Prediction** → **Sequence/Tree/Graph Decoder**
## Performance of Built-in NLP Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Dataset</th>
<th>GNN Model</th>
<th>Graph construction</th>
<th>Evaluation</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text classification</td>
<td>TRECT</td>
<td>GAT</td>
<td>Dependency</td>
<td>Accuracy</td>
<td>0.948</td>
</tr>
<tr>
<td></td>
<td>CAirline CNSST</td>
<td></td>
<td></td>
<td></td>
<td>0.769</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.538</td>
</tr>
<tr>
<td>Semantic Parsing</td>
<td>JOBS</td>
<td>SAGE</td>
<td>Constituency</td>
<td>Execution accuracy</td>
<td>0.936</td>
</tr>
<tr>
<td>Question generation</td>
<td>SQuAD</td>
<td>GGNN</td>
<td>Dependency</td>
<td>BLEU-4</td>
<td>0.15175</td>
</tr>
<tr>
<td>Machine translation</td>
<td>IWSLT14</td>
<td>GCN</td>
<td>Dynamic</td>
<td>BLEU-4</td>
<td>0.3212</td>
</tr>
<tr>
<td>Summarization</td>
<td>CNN(30k)</td>
<td>GCN</td>
<td>Dependency</td>
<td>ROUGE-1</td>
<td>26.4</td>
</tr>
<tr>
<td>Knowledge graph completion</td>
<td>Kinship</td>
<td>GCN</td>
<td>Dependency</td>
<td>MRR</td>
<td>82.4</td>
</tr>
<tr>
<td>Math word problem</td>
<td>MAWPS MATHQA</td>
<td>SAGE</td>
<td>Dynamic</td>
<td>Solution accuracy Exact match</td>
<td>76.4 61.07</td>
</tr>
</tbody>
</table>
Demo 1: Building a Text Classification Application

1) git clone https://github.com/graph4ai/graph4nlp_demo
2) follow Get Started instructions in README
Demo 1: Building a Text Classification Application

```python
def forward(self, graph_list, tgt=None, require_loss=True):
    # build graph topology
    batch_gd = self.graph_topology(graph_list)

    # run GNN encoder
    self.gnn(batch_gd)

    # run graph classifier
    self.clf(batch_gd)
    logits = batch_gd.graph_attributes['logits']

    if require_loss:
        loss = self.loss(logits, tgt)
        return logits, loss
    else:
        return logits
```

https://github.com/graph4ai/graph4nlp_demo/tree/main/NAACL2021_demo
Demo 1: Building a Text Classification Application

```python
self.graph_topology = DependencyBasedGraphConstruction(
    embedding_style=embedding_style,
    vocab=vocab.in_word_vocab,
    hidden_size=config['num_hidden'],
    word_dropout=config['word_dropout'],
    rnn_dropout=config['rnn_dropout'],
    fix_word_emb=not config['no_fix_word_emb'],
    fix_bert_emb=not config.get('no_fix_bert_emb', False))
```

Graph construction API, various built-in options, can be customized

https://github.com/graph4ai/graph4nlp_demo/tree/main/NAACL2021_demo
Demo 1: Building a Text Classification Application

```python
self.gnn = GraphSAGE(config['gnn_num_layers'],
                     config['num_hidden'],
                     config['num_hidden'],
                     config['num_hidden'],
                     config['graphsage_aggregation_type'],
                     direction_option=config['gnn_direction_option'],
                     feat_drop=config['gnn_dropout'],
                     bias=True,
                     norm=None,
                     activation=F.relu,
                     use_edge_weight=use_edge_weight)
```

GNN API, various built-in options, can be customized

https://github.com/graph4ai/graph4nlp_demo/tree/main/NAACL2021_demo
Demo 1: Building a Text Classification Application

```python
self.clf = FeedForwardNN(2 * config['num_hidden'] \n    if config['gnn_direction_option'] == 'bi_sep' \n    else config['num_hidden'],
    config['num_classes'],
    [config['num_hidden']],
    graph_pool_type=config['graph_pooling'],
    dim=config['num_hidden'],
    use_linear_proj=config['max_pool_linear_proj'])
```

Prediction API, various built-in options, can be customized

https://github.com/graph4ai/graph4nlp_demo/tree/main/NAACL2021_demo
Demo 1: Building a Text Classification Application

Dataset API, various built-in options, can be customized

```python
dataset = TrecDataset(root_dir=self.config.get('root_dir', self.config['root_data_dir']),
                      pretrained_word_emb_name=self.config.get('pretrained_word_emb_name', '840B'),
                      merge_strategy=merge_strategy,
                      seed=self.config['seed'],
                      thread_number=4,
                      port=9000,
                      timeout=15000,
                      word_emb_size=300,
                      graph_type=graph_type,
                      topology_builder=topology_builder,
                      topology_subdir=topology_subdir,
                      dynamic_graph_type=self.config['graph_type'] if \
                          self.config['graph_type'] in ('node_emb', 'node_emb_refined') else None,
                      dynamic_init_topology_builder=dynamic_init_topology_builder,
                      dynamic_init_topology_aux_args={'dummy_param': 0})
```
Demo 2: Building a Semantic Parsing Application

1) git clone https://github.com/graph4ai/graph4nlp_demo
2) follow Get Started instructions in README
Demo 2: Building a Semantic Parsing Application

```python
def _build_model(self):
    self.model = Graph2Seq.from_args(self.opt, self.vocabulary).to(self.device)
```

https://github.com/graph4ai/graph4nlp_demo/tree/main/NAACL2021_demo
Demo 2: Building a Semantic Parsing Application

```python
dataset = JobsDataset(root_dir=self.opt["graph_construction_args"],
                      graph_construction_share=
                      self.opt["graph_construction_private"],
                      merge_strategy=
                      self.opt["graph_construction_args"],
                      val_split_ratio=
                      self.opt["val_split_ratio"],
                      merge_strategy=
                      self.opt["graph_construction_args"],
                      edge_strategy=
                      self.opt["graph_construction_args"],
                      seed=self.opt["seed"],
                      word_emb_size=self.opt["word_emb_size"],
                      share_vocab=self.opt["graph_construction_args"],
                      graph_type=graph_type,
                      topology_builder=topology_builder,
                      topology_subdir=self.opt["graph_construction_args"],
                      thread_number=self.opt["graph_construction_args"],
                      dynamic_graph_type=self.opt["graph_construction_args"],
                      dynamic_init_topology_builder=dynamic_init_topology_builder,
                      dynamic_init_topology_aux_args= None)
```

Dataset API, various built-in options, can be customized

https://github.com/graph4ai/graph4nlp_demo/tree/main/NAACL2021_demo
DLG4NLP: Future Directions and Conclusions
Future Directions

- The Rise of GNN + NLP

#ICLR2020 submissions on graph neural networks, NLP and robustness have the greatest growth. @iclr_conf @openreviewnet

- Graph Construction for NLP
  - Dynamic graph construction are largely underexplored!
  - How to effectively combine advantages of static graph and dynamic graph?
  - How to construct heterogeneous dynamic graph?
  - How to make dynamic graph construction itself scalable?
Future Directions

• **Scaling GNNs to Large Graphs**
  • Most existing multi-relational or heterogeneous GNNs will have scalability issues when applied to large graphs in NLP such as KGs (> 1m)

• **GNNs + Transformer in NLP**
  • How to effectively combine the advantages of GNNs and Transformer?
  • Is graph transformer the best way to utilize?

• **Pretraining GNNs for NLP**
  • Information Retrieval/ Search
Future Directions

• **Graph-to-graph Learning in NLP**
  • How to effectively develop Graph-to-Graph models for solving graph transformation problem in NLP (i.e. information extraction)?

• **Joint Text and KG Reasoning in NLP**
  • Joint text and KG reasoning is less explored although GNNs for multi-hop reasoning gains popularity

• **Incorporate Source and Context into Knowledge Graph Construction and Verification**
Conclusions

- Deep Learning on Graphs for NLP is a fast-growing area today!
- Since graph can naturally encode complex information, it could bridge a gap by combining both empirical domain knowledges and the power of deep learning.
- For a NLP task,
  - how to convert text sequence into the best graph (directed, multi-relation, heterogeneous)
  - how to determine proper graph representation learning technique?
- Our Graph4NLP library aims to make easy use of GNNs for NLP:
  - Code: https://github.com/graph4ai/graph4nlp
  - Demo: https://github.com/graph4ai/graph4nlp_demo
  - Github literature list: https://github.com/graph4ai/graph4nlp_literature