GLoMo: Unsupervisedly Learned Relational Graphs as Transferable Representations

Zhilin Yang, Jake Zhao, Bhuwan Dhingra, Kaiming He, William W. Cohen, Ruslan Salakhutdinov, Yann LeCun

Presenter: Arshdeep Sekhon

https://qdata.github.io/deep2Read

Motivation

- CNNs and RNNs are tailored to grids or sequences
- To compensate for not being able to model graphical representations: use deep model expressiveness
- transfer learning standard approach: train on large dataset and then apply to extract features

Real Data has graph structures

- Real data has graph structures
  - parse trees, knowledge graphs, coreference resolution
  - same across all language tasks
- (-) expensive to acquire: human curated
- automatically-induced structures are mostly limited to one task (example NRI, Attention is all you need, Non-local neural networks)

GLOMO Objectives

- instead of unary feature learning : learn graphical features
- to learn *l*atent relational graphs : input units are nodes (example words in a sentence)

GLOMO

- Graphs from Low-level unit MOdeling
- train a *n*eural network that learns latent graphs from large scale unsupervised data
- transfer this *n*eural network to downstream task: outputs new graphs

Why?

separate the features that represent the semantic meaning of each unit and the graphs that reflect how the units may interact. Ideally, the graphs capture task-independent structures underlying the data, and thus become applicable to different sets of features.

Outline

Input

Given a sample/sequence:

$$(x_1, x_2, \ldots, x_T) \tag{1}$$

- Learn G a $T \times T$, matrix where $G_{i,j}$ represents affinity between x_i and x_j

Model

- Graph Predictor: $G = g(x)$, $G \in R^{L \times T \times T}$
- L the number of layers that produce graphs
- Feature Predictor $f(G, x)$
- During test time, in a downstream task, extract G using Graph predictor

Outline

Unsupervised Learning: Graph Predictor

- Made of two CNNs: key CNN and a query CNN
- input to both CNNs is x
- output of key CNN : $(k_1, \ldots, k_T)$
- output of query CNN : $q_1, \ldots, q_T$
- Compute Graph using:

$$G_{ij}^{l} = \frac{ReLU(k_i^{lT} q_j^{l} + b)^2}{\sum_i' ReLU(k_i'^{lT} q_j^{l} + b)^2} \tag{2}$$

- same as attention weights other than square and ReLU
- *we use ReLUs to enforce sparsity and the square operations to stabilize training.Moreover, we employ convolutional networks to let the graphs G be aware of the local order of the input and context, up to the size of each unit's receptive field.*

Outline

Feature Predictor

- $\boldsymbol{f}_t^l = v((\sum_j \boldsymbol{G}_{jt}^l \boldsymbol{f}_{jt}^{l-1}), \boldsymbol{f}_t^{l-1})$
- at $t = 0$, $f_t^0 = x_t$ input embeddings
- v can be MLP or GRU Cell

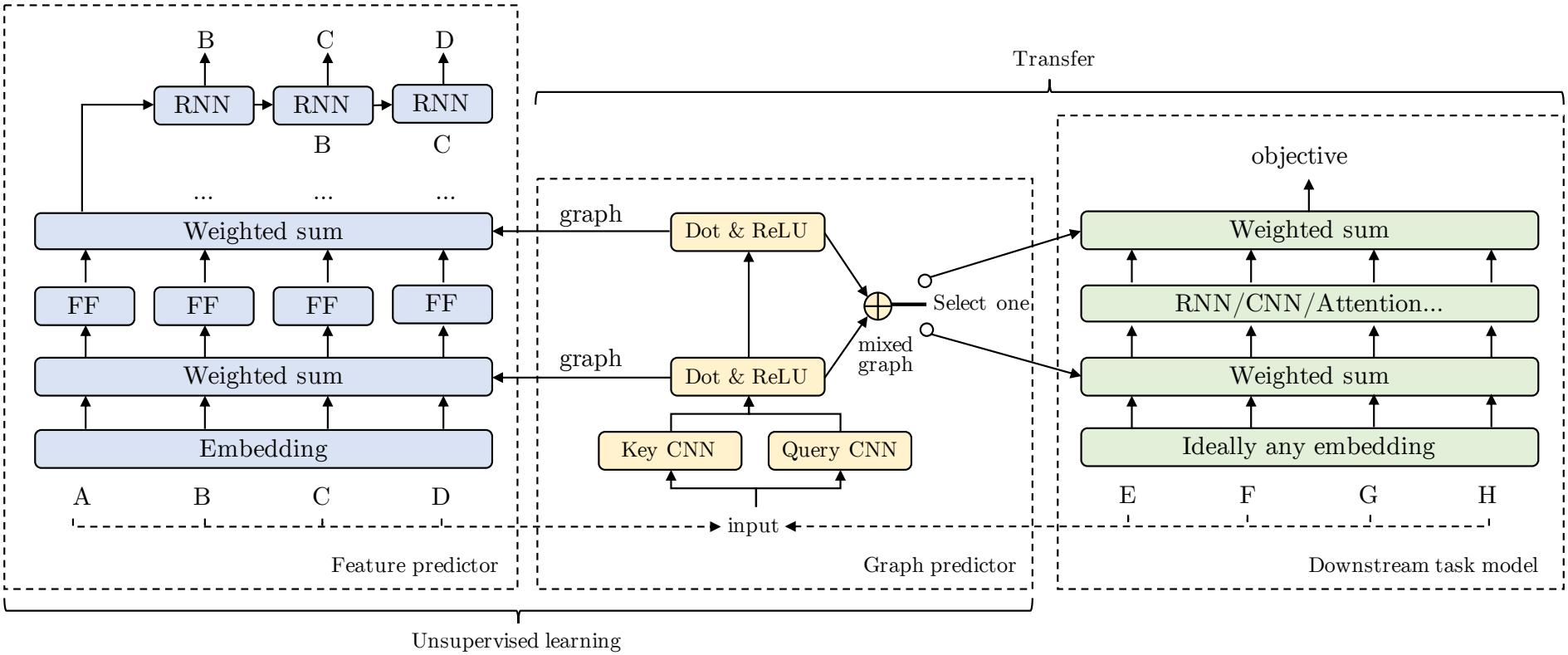
Feature Predictor

- Use $\boldsymbol{f}_t^L$ to initialize hidden state
- predict units following x_t
- Objective function:

$$max \sum_t logP(x_{t+1}, \ldots, x_{t+D} | x_t, \boldsymbol{f}_t^L) \tag{3}$$

- mask the convolutional filters and the graph G (see Eq. 1) in the network g to prevent the network from accessing the future

GLOMO


B
C
D
RNN
RNN
RNN
B
C
...
...
...
Weighted sum
FF
FF
FF
FF
Weighted sum
Embedding
A
B
C
D
Feature predictor
graph
Dot & ReLU
graph
Dot & ReLU
Select one
mixed graph
Key CNN
Query CNN
input
Graph predictor
Transfer
objective
Weighted sum
RNN/CNN/Attention...
Weighted sum
Ideally any embedding
E
F
G
H
Downstream task model
Unsupervised learning

Differences with self-attention and predictive unsupervised learning

- Decoupling graphs and features: separate networks g and f
- Sparsity: a squared ReLU activation, most NLP graphs are sparse

  We believe sparse structures reduce noise and are more transferable.
- Hierarchical graph representations: multiple layers of graphs, which allows us to model hierarchical structures in the data.

Latent Graph Transfer

- Say predictor takes in features $\boldsymbol{H} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_T)$
- produces $(\boldsymbol{h'}_1, \ldots, \boldsymbol{h'}_T)$
- Get graphs from Graph predictor from all l layers $\boldsymbol{G}$
- $\Lambda^l = \prod_{i=1}^{l} \boldsymbol{G}^l$
-

$$\boldsymbol{M} = \sum_{l=1}^{L} m_g^l \boldsymbol{G}^l + \sum_{l=1}^{L} m_\Lambda^l \Lambda^l \tag{4}$$

- $\sum_l (m_\Lambda^l + m_g^l) = 1$
- use $\boldsymbol{HM}$ as input

Datasets

- Question Answering: SQuAD dataset
- Natural Language Inference: Multi-Genre NLI corpus: sentence pairs annotated with textual entailment information
- Sentiment Analysis: Movie Review from IMDB
- Transfer Setting: Wikipedia (700 million)

Results on Natural Language Tasks

Transfer method	SQuAD GloVe *EM*	SQuAD GloVe *F1*	SQuAD ELMo *EM*	SQuAD ELMo *F1*	IMDB GloVe *Accuracy*	MNLI GloVe *matched*	MNLI GloVe *mism.*
transfer feature only (baseline)	69.33	78.73	74.75	82.95	88.51	77.14	77.40
GLoMo on embeddings	70.84	79.90	**76.00**	**84.13**	**89.16**	**78.32**	**78.00**
GLoMo on RNN states	**70.95**	**79.95**	75.59	83.62	-	-	-

Ablation Study

Table 2: Ablation study.

Method	SQuAD GloVe		SQuAD ELMo		IMDB GloVe	MNLI GloVe	
	EM	*F1*	*EM*	*F1*	*Accuracy*	*matched*	*mism.*
GLoMo	**70.84**	**79.90**	**76.00**	**84.13**	**89.16**	**78.32**	78.00
- decouple	70.45	79.56	75.89	83.79	-	-	-
- sparse	70.13	79.34	75.61	83.89	88.96	78.07	77.75
- hierarchical	69.92	79.23	75.70	83.72	88.71	77.87	77.85
- unit-level	69.23	78.66	74.84	83.37	88.49	77.58	**78.05**
- sequence	69.92	79.29	75.50	83.70	88.96	78.11	77.76
uniform graph	69.48	78.82	75.14	83.28	88.57	77.26	77.50

Image Classification

Method / Base-model	ResNet-18	ResNet-34
baseline	90.93±0.33	91.42±0.17
GLoMo	**91.55±0.23**	**91.70±0.09**
ablation: uniform graph	91.07±0.24	-

Conclusions

- transferable graph structures
- transfer graph learner module
- learn graph from large datasets and transfer to other tasks.