最先端NLP勉強会

Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling

Diego Marcheggiani　Ivan Titov
ILLC, University of Amsterdam
ILCC, School of Informatics, University of Edinburgh

In Proc. of EMNLP 2017

豊田工業大学　知能数理研　修士2年　辻村有輝

概要

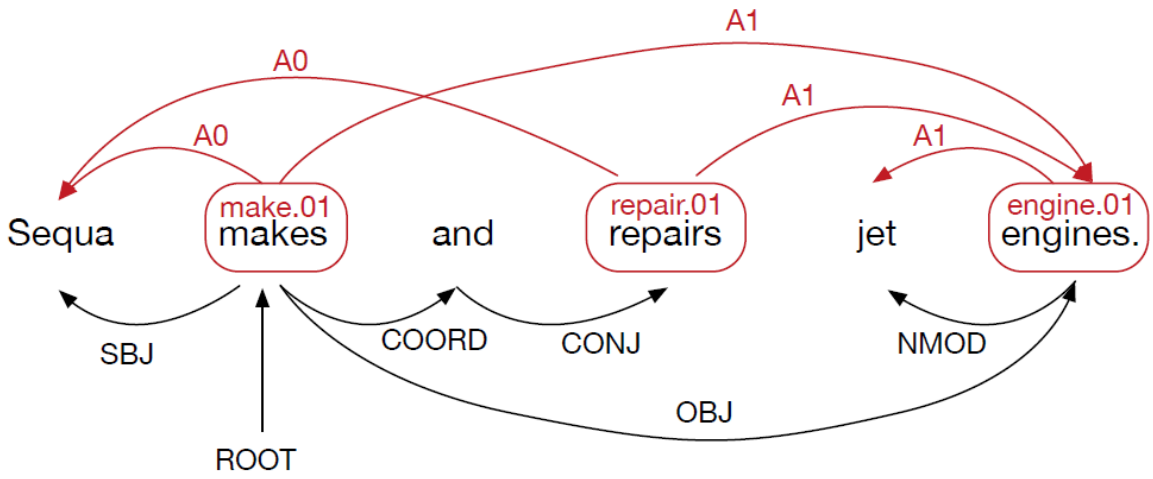
- 述語項構造解析 (**S**emantic **R**ole **L**abeling, **SRL**)
- **G**raph **C**onvolutional **N**etworks（**GCN**）を使用
- モデルは多層BiLSTM+多層GCN+出力層

- 多くは第一著者の別論文（CoNLL-2017）に基づく

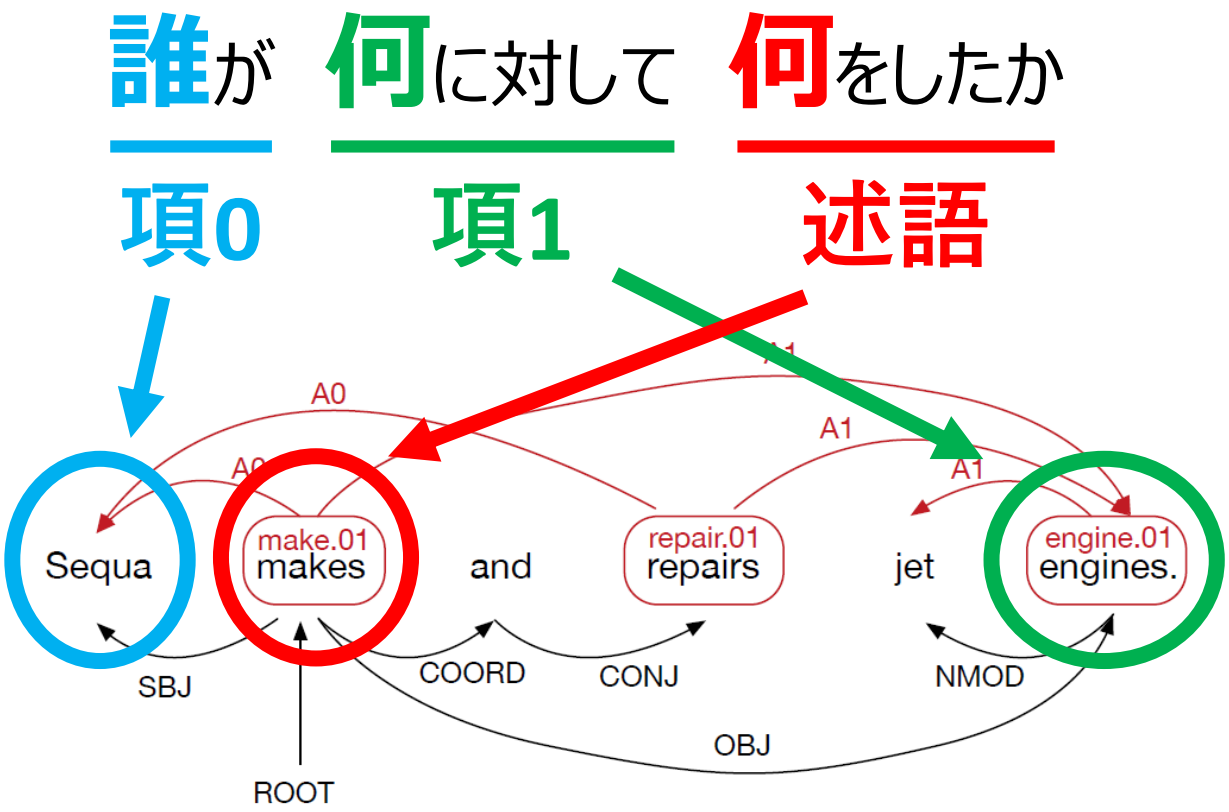
述語項構造解析

誰が　**何**に対して　**何**をしたか

項0　　**項1**　　**述語**



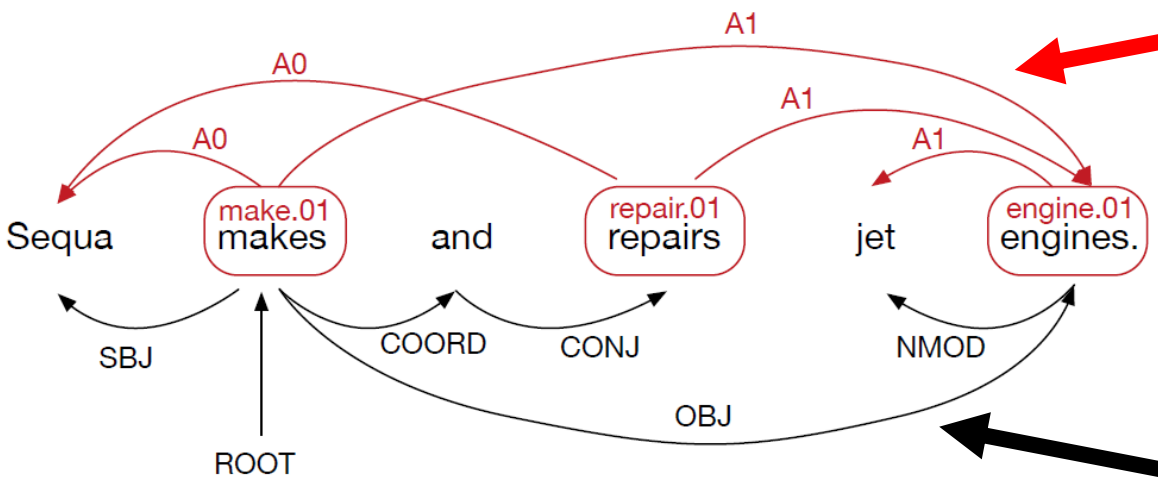
述語項構造解析


誰が
何に対して
何をしたか
項0
項1
述語
A0
A1
A0
A1
Sequa
make.01
makes
and
repair.01
repairs
jet
engine.01
engines.
SBJ
ROOT
COORD
CONJ
NMOD
OBJ

述語項構造解析（データセットに関して）

- この論文で利用されるCoNLL-2009のデータセットでは文中でどれが述語かについてとPOSタグは与えられている
- この論文では英語と中国語を対象にする
- 係り受け木は別システムorデータセット付属のシステムであらかじめ解析

述語項構造解析



述語項構造

係り受け関係

係り受け関係（構文）と述語項構造は
深く結びついている

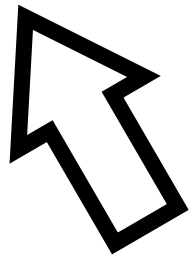
➡ 構文情報を使いたい

述語項構造解析

- 伝統的には構文情報を利用したモデルが用いられてきた
- 最近では構文情報を使わない複数のBiLSTMを
  スタックしたモデルが高性能

述語項構造解析

- 伝統的には構文情報を利用したモデルが用いられてきた
- 最近では構文情報を使わない複数のBiLSTMを
  スタックしたモデルが高性能



簡単で効果的な構文情報を利用する方法が
あまり無いために使われなかった

➡ **G**raph **C**onvolutional **N**etwork (**GCN**)

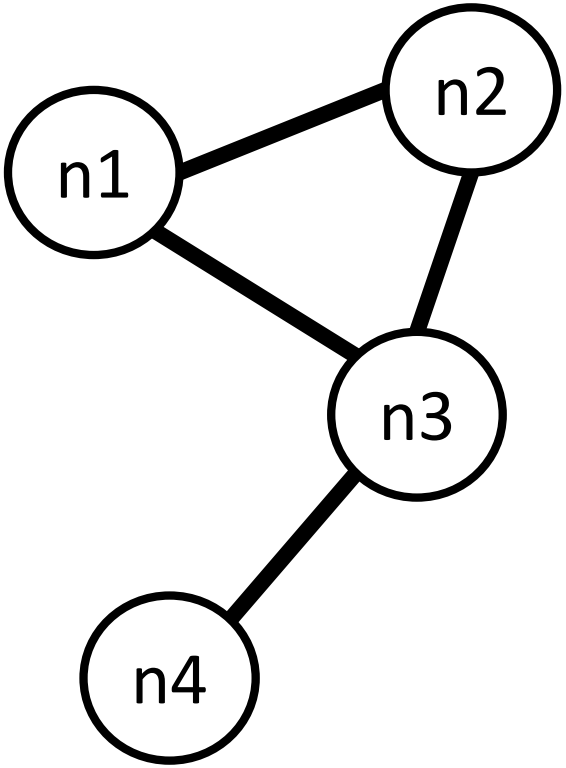
Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル

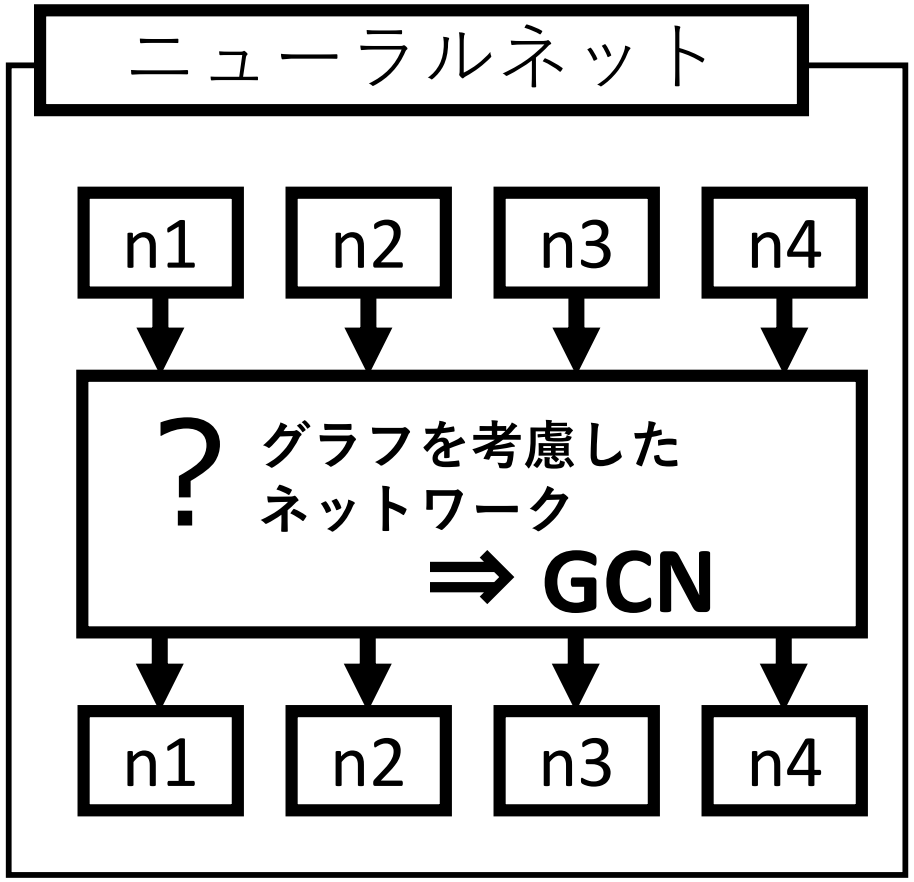
- Recursive NNと違い閉ループを持つ構造など
  木構造以外にも適用可能なモデル
  - ただし今回は係り受け木
- 1層のGCNでは1近傍のみを見る ← 近似のせい
- 遠い関係のためには多層積み重ねる必要がある
  - ただし今回は最終的に1層

Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル

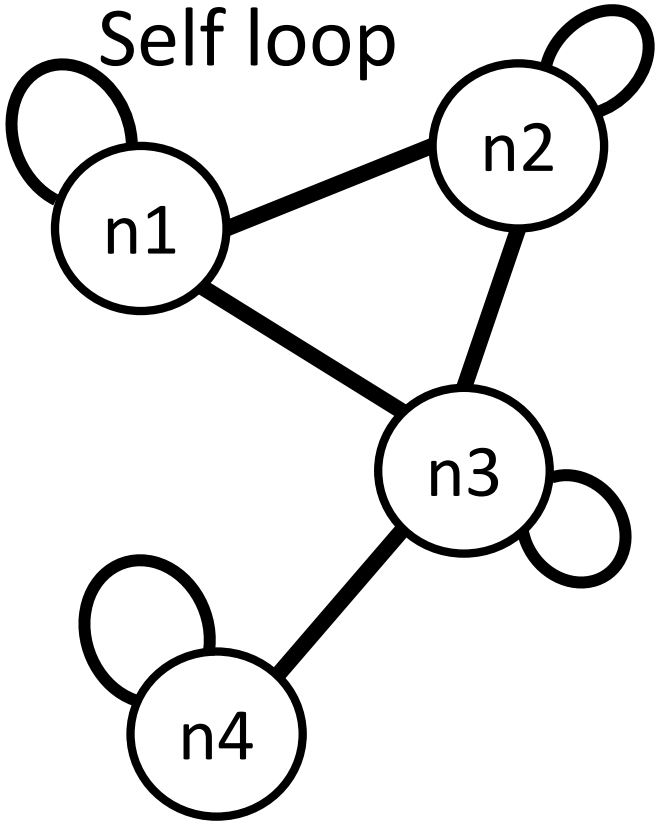


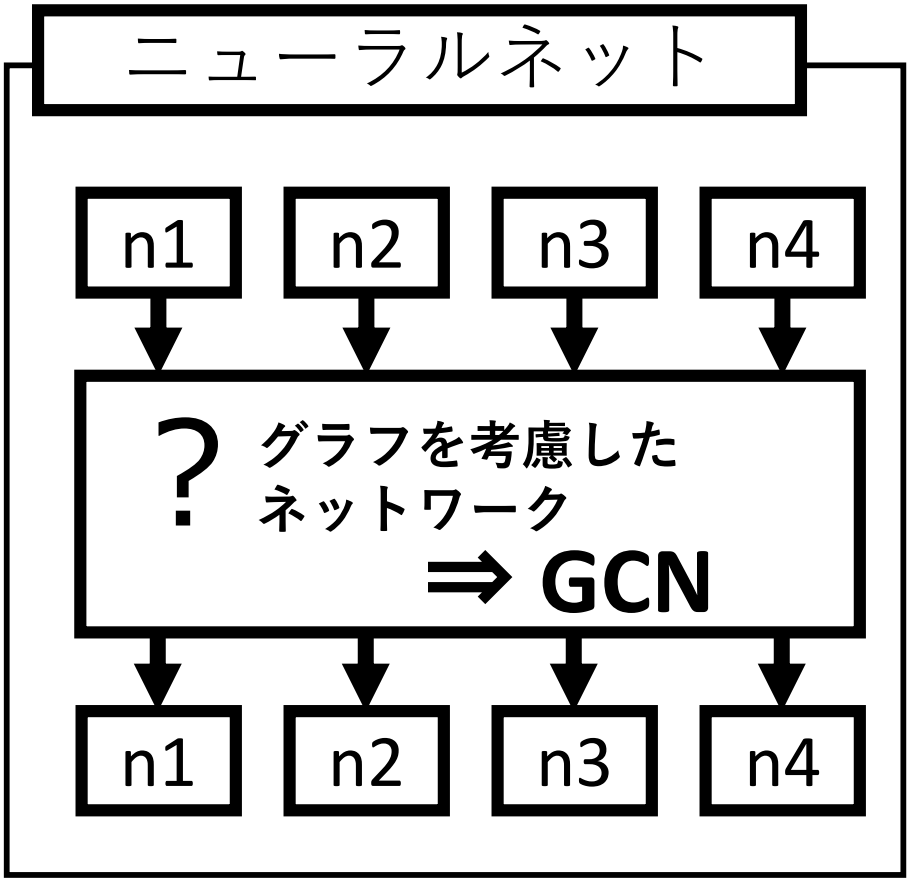
無向グラフを想定



Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル

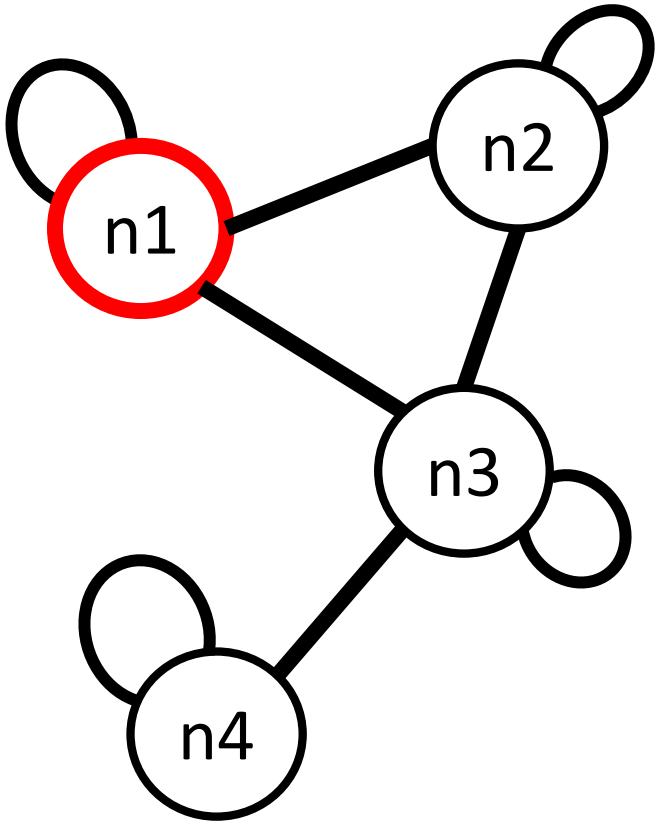




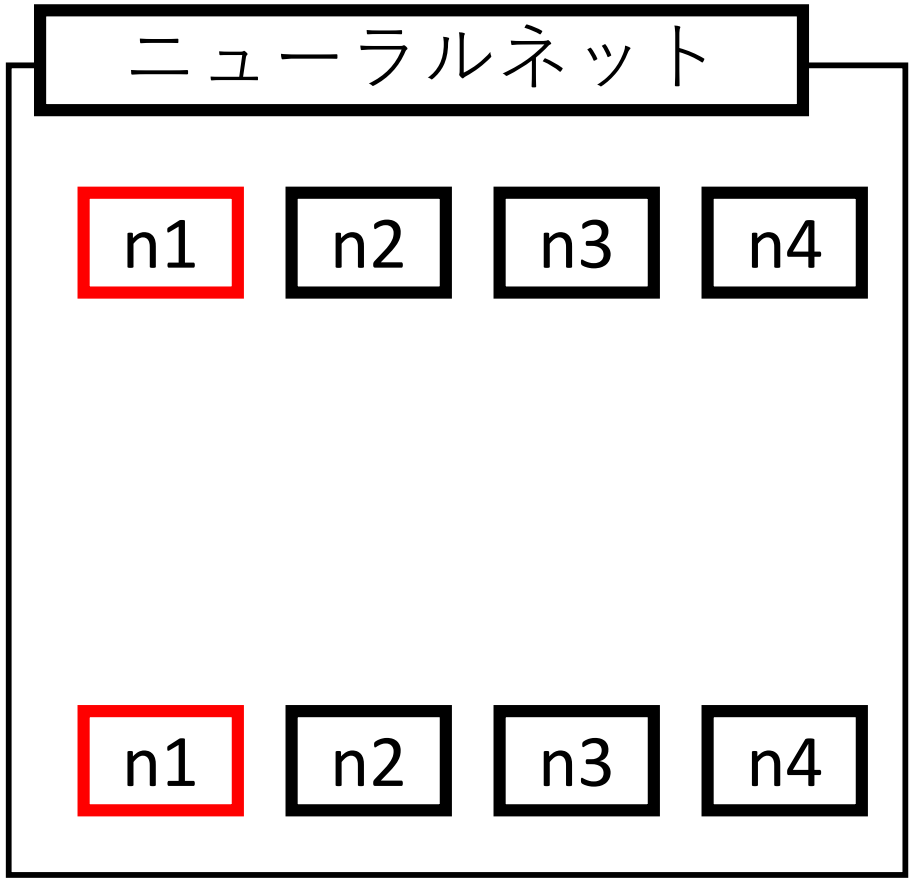
Self loopがあるとする

Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル

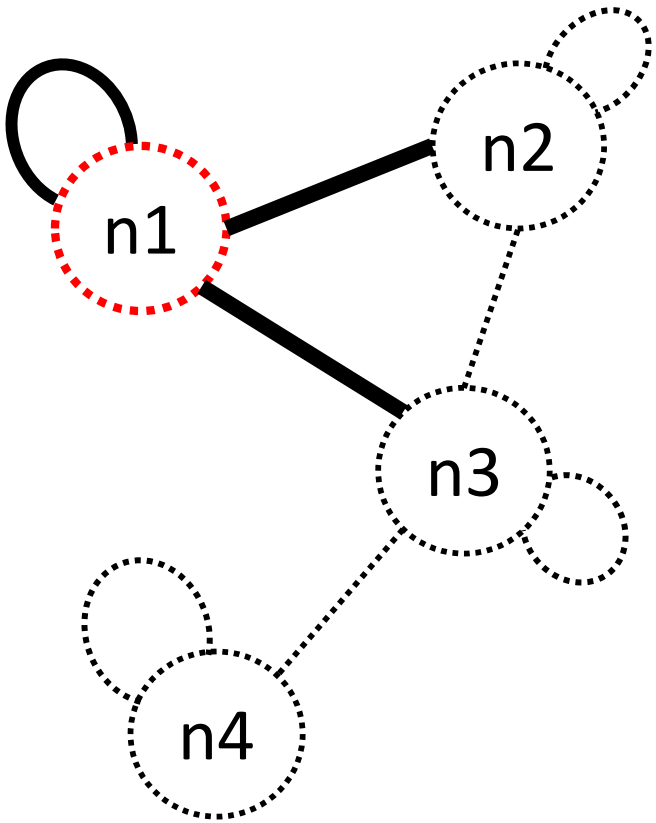


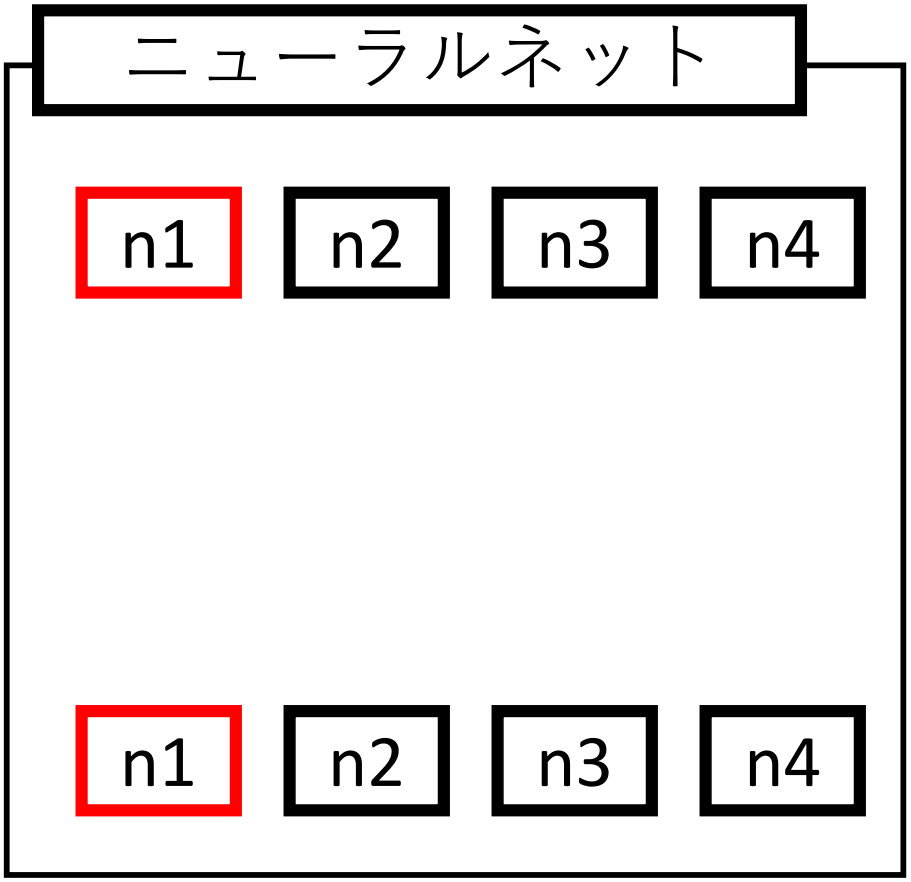
あるノードに注目



Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル

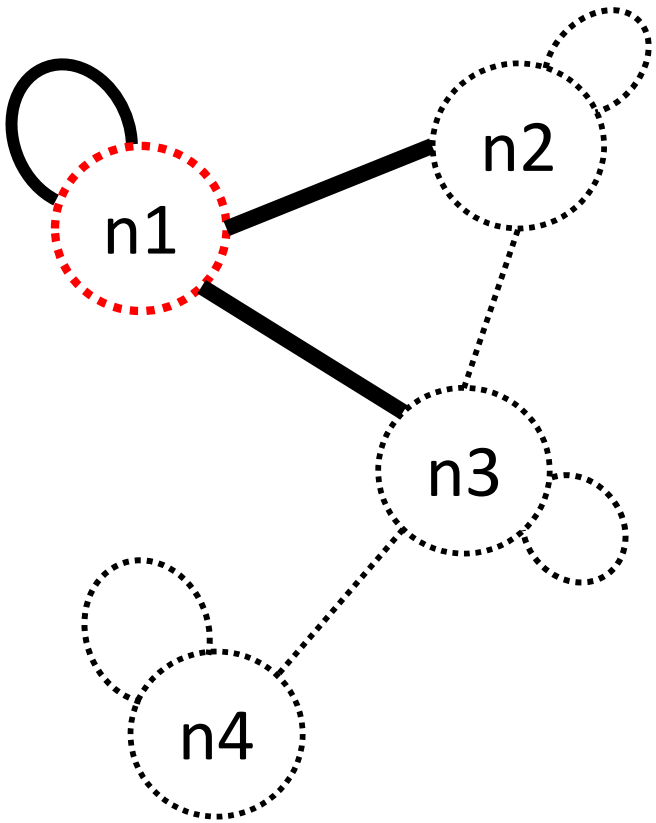


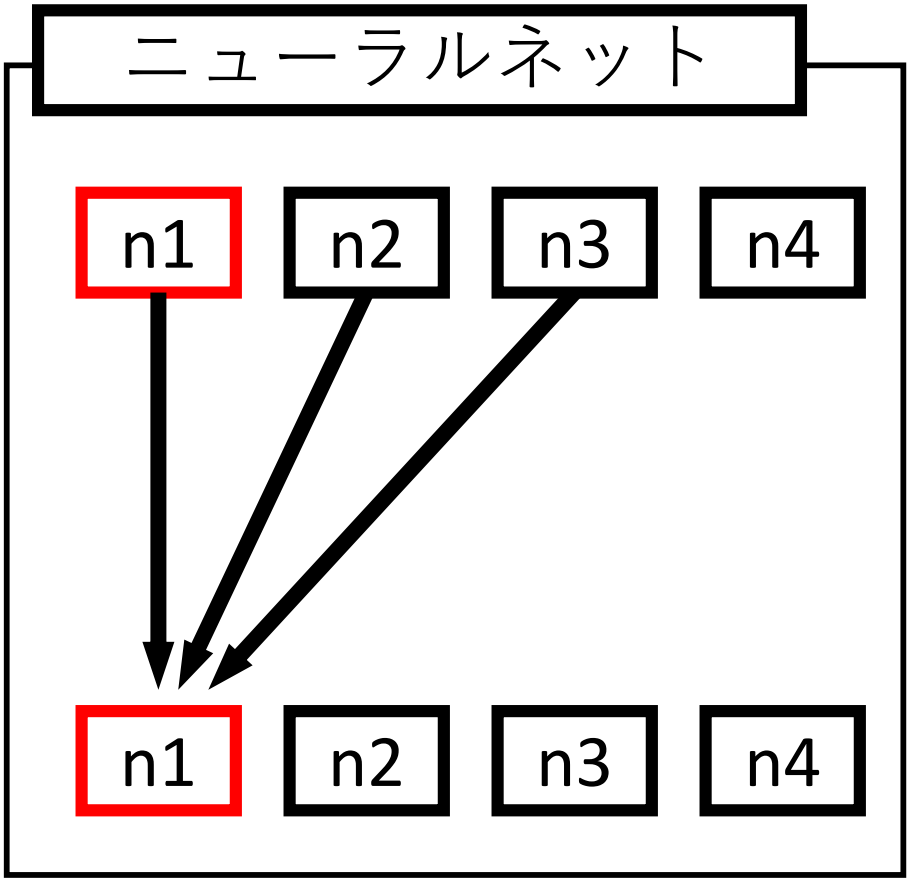


エッジの通りに

Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル

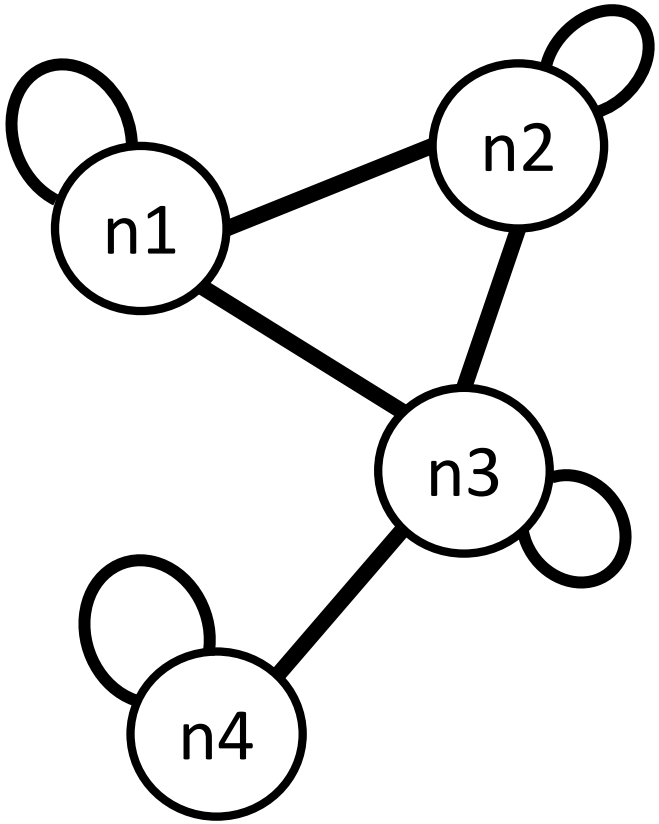


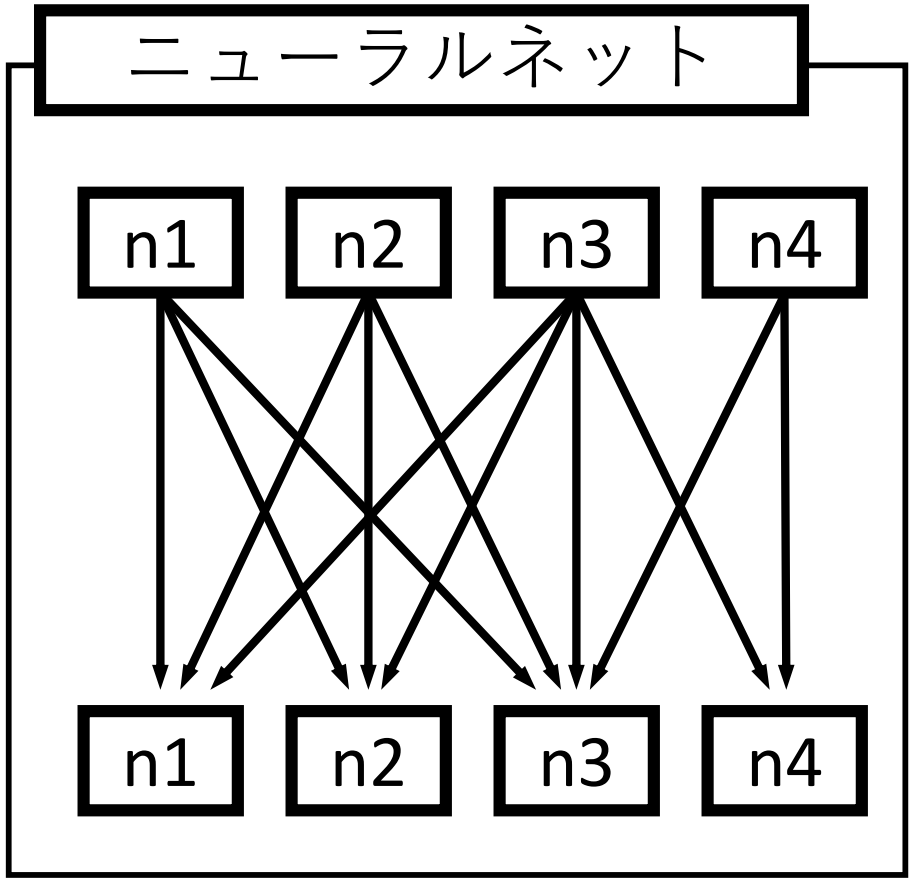


出力に向けて繋げる

Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル

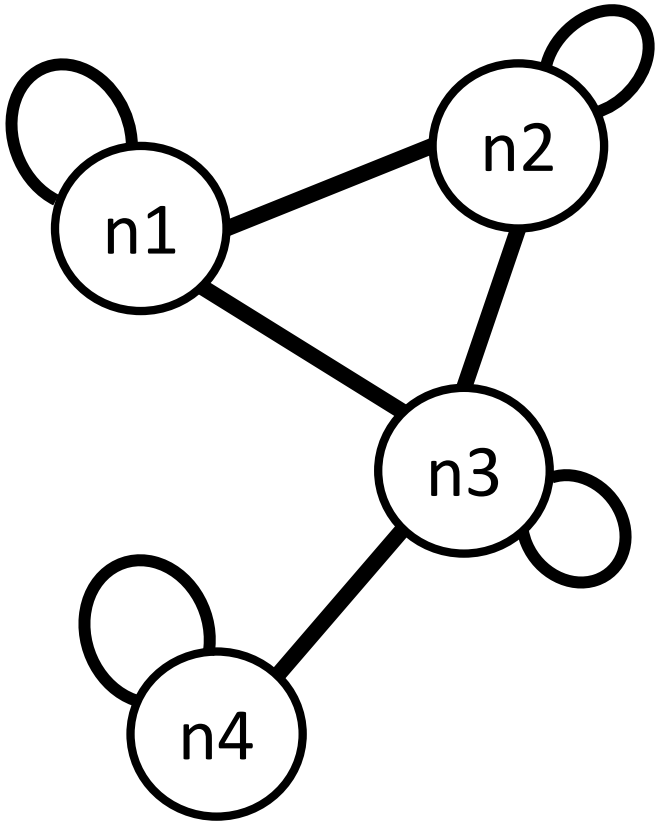


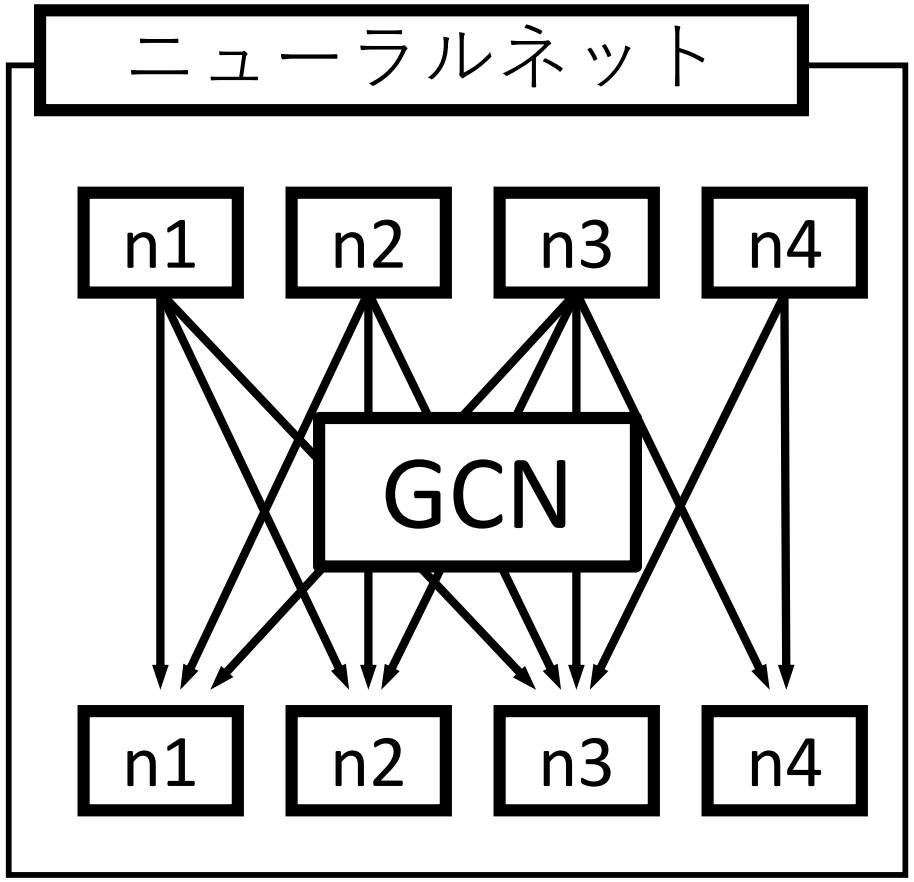


他も同じように繋げる

Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル





(最後にReLUをかける)

Graph **C**onvolutional **N**etworks

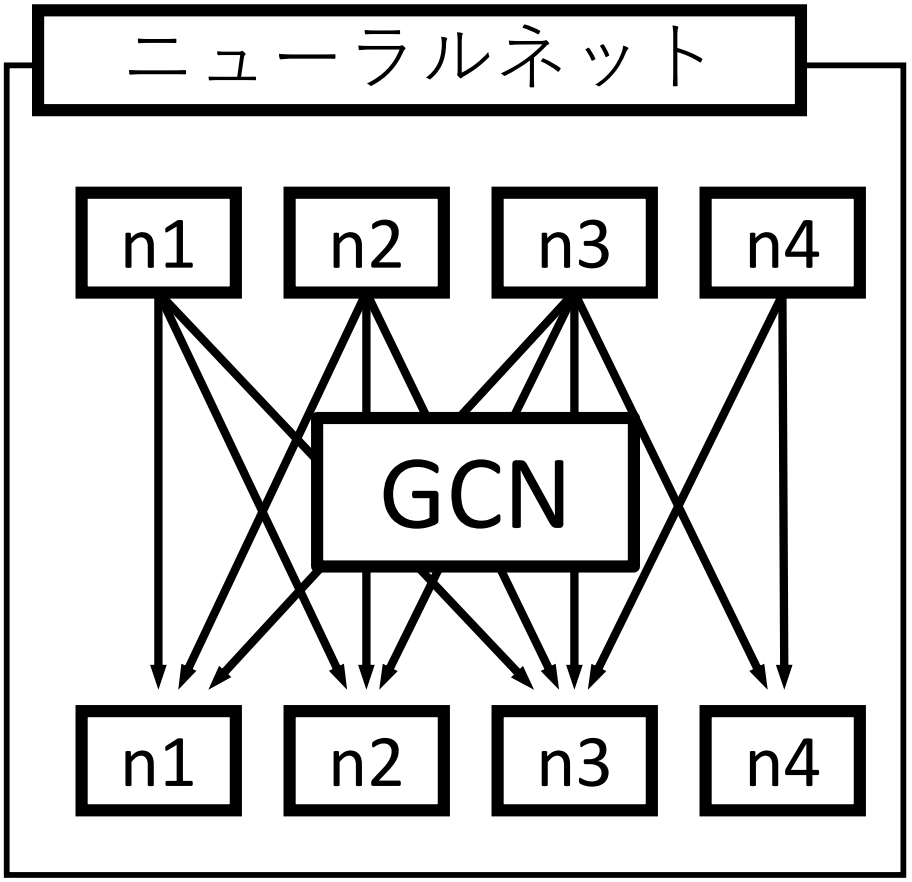
グラフ用のニューラルネットワークモデル

$$h_v = ReLU\left(\sum_{u \in N(v)} (Wx_u + b)\right)$$

式は

1. グラフ上のシグナル（ベクトル）$\boldsymbol{x}$に対してフーリエ変換を考える
2. $\boldsymbol{x} * \theta = FT^{-1}\left(FT(\boldsymbol{x}) \odot FT(\theta)\right)$で畳み込みの式を得る
3. 近似

で得られる



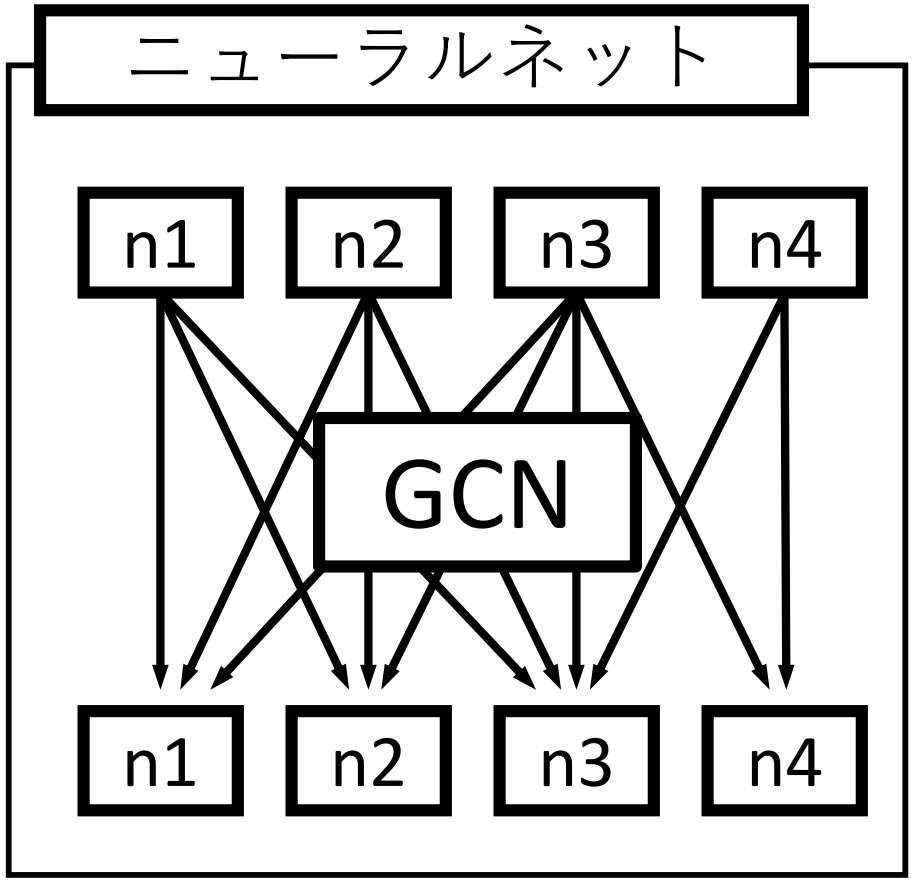
Graph **C**onvolutional **N**etworks

グラフ用のニューラルネットワークモデル

$$h_v = ReLU\left(\sum_{u \in N(v)} (W x_u + b)\right)$$

- 木構造以外にも適用可能
- 1層で1近傍を見る
- **そのままでは無向グラフ用**

➡係り受け木で使えるよう
いくつか変更をする



係り受け木のためのGCN-1/3

- 元の式は無向グラフ前提

$$h_v = ReLU\left(\sum_{u \in N(v)} (W x_u + b)\right)$$

- 構文木は有向グラフでエッジにも種類がある

➡ *W,b*をエッジ依存に

$$h_v = ReLU\left(\sum_{u \in N(v)} \left(W_{L(u,v)} x_u + b_{L(u,v)}\right)\right)$$

$L(u,v)$：ノード*u,v*間のエッジを表す関数

係り受け木のためのGCN-2/3

*W,b*をエッジ依存にする

$$h_v = ReLU\left(\sum_{u \in N(v)} \left(W_{L(u,v)} x_u + b_{L(u,v)}\right)\right)$$

このままでは係り受けの種類分 (大量) の*W*が増える

➡ $W_{L(u,v)} = W_{dir(u,v)}$

“パスが順方向” or ”パスが逆方向” or ”Self loop”の
3種類のみにする

※*b*に関しては許容（*b*によってエッジ種類の考慮を期待）

係り受け木のためのGCN-3/3

今のままでは子ノードからの情報は等しく受け入れてしまう

$$h_v = ReLU\left(\sum_{u \in N(v)} \left(W_{dir(u,v)} x_u + b_{L(u,v)}\right)\right)$$

x_u：子のベクトル

➡ 親依存のゲートを追加する

$$h_v = ReLU\left(\sum_{u \in N(v)} g_{v,u} \times \left(W_{dir(u,v)} x_u + b_{L(u,v)}\right)\right)$$

$$g_{v,u} = \sigma\left(x_v \cdot v_{dir(v,u)} + k_{L(v,u)}\right)$$

x_v：親

$v_{dir(v,u)}$：重みベクトル

$g_{v,u}$：スカラ

係り受け木のためのGCN

$$h_v = ReLU\left(\sum_{u \in N(v)} g_{v,u} \times \left(W_{dir(u,v)} x_u + b_{L(u,v)}\right)\right)$$

$$g_{v,u} = \sigma\left(x_v \cdot v_{dir(v,u)} + k_{L(v,u)}\right)$$

- *W,b*をエッジ依存に
- 増えすぎた*W*をエッジの向きのみに依存するように
- ゲートによって子を取捨選択

モデル

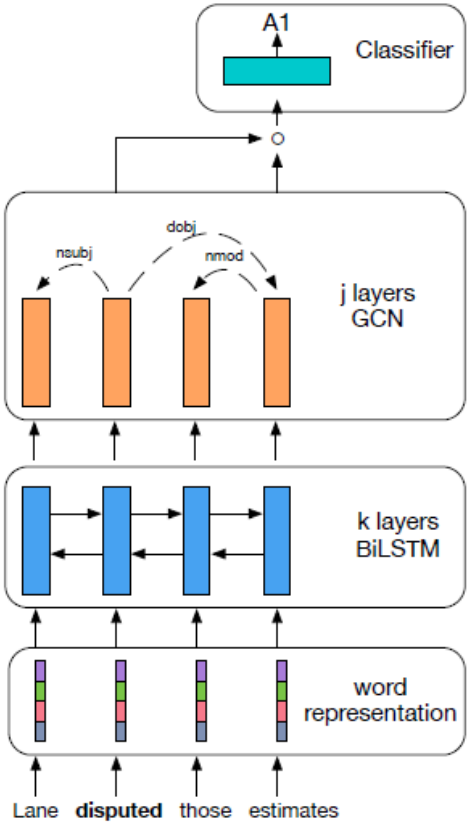


Figure 3: Predicting an argument and its label with an LSTM + GCN encoder.

- 入力層
- 多層BiLSTM
- 多層GCN
- 出力層

モデル

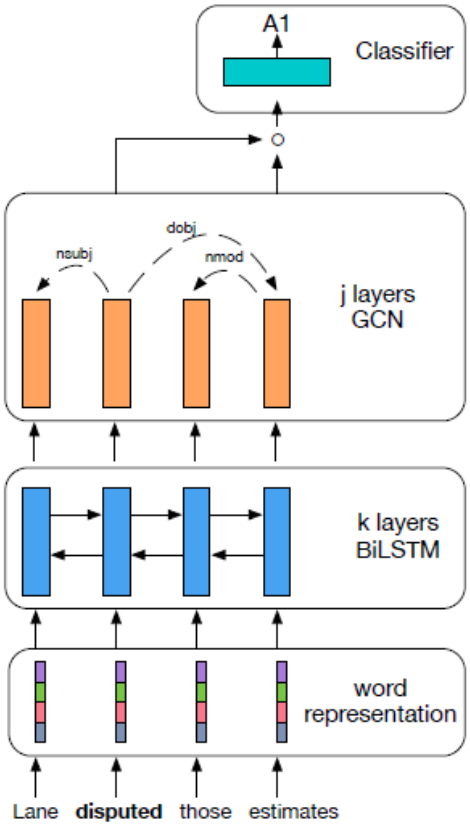


Figure 3: Predicting an argument and its label with an LSTM + GCN encoder.

・入力層
- 事前学習済み単語ベクトル
  (fine tuningしない)
- ランダム初期化した単語ベクトル
- ランダム初期化したPOSベクトル
- ランダム初期化したlemmaベクトル

これら4種を連結したもの

さらにその単語が注目中の述語かどうかの0/1な素性が追加で連結されている

モデル

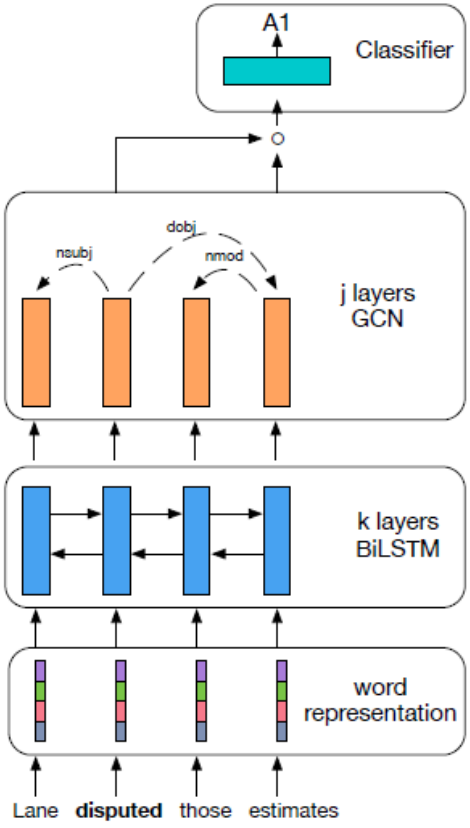


Figure 3: Predicting an argument and its label with an LSTM + GCN encoder.

・出力層

$$p(r|t_i, t_p, l) \propto \exp(W_{l,r}(t_i \circ t_p)), \quad (5)$$

$$W_{l,r} = ReLU(U(q_l \circ q_r)), \quad (6)$$

t_i ：i番目のGCNの出力

U ：重み行列

q_l ：lemmaの埋め込み

q_r ：述語の意味役割の埋め込み

モデル

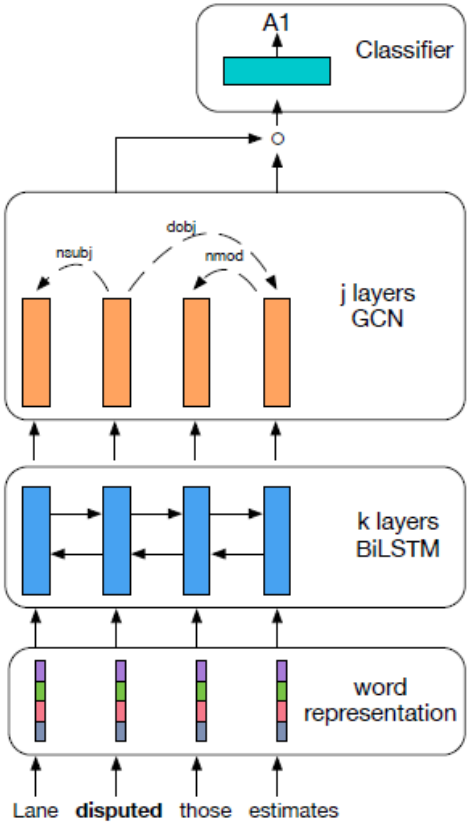


Figure 3: Predicting an argument and its label with an LSTM + GCN encoder.

- 入力層
- 多層BiLSTM（3層）
- 多層GCN（1層）
- 出力層

実験結果

GCNの総数による比較

System (English)	P	R	F_1
LSTMs	84.3	81.1	82.7
LSTMs + GCNs (K=1)	85.2	81.6	83.3
LSTMs + GCNs (K=2)	84.1	81.4	82.7
LSTMs + GCNs (K=1), no gates	84.7	81.4	83.0
GCNs (no LSTMs), K=1	79.9	70.4	74.9
GCNs (no LSTMs), K=2	83.4	74.6	78.7
GCNs (no LSTMs), K=3	83.6	75.8	79.5
GCNs (no LSTMs), K=4	82.7	76.0	79.2

Table 1: SRL results without predicate disambiguation on the English development set.

System (Chinese)	P	R	F_1
LSTMs	78.3	72.3	75.2
LSTMs + GCNs (K=1)	79.9	74.4	77.1
LSTMs + GCNs (K=2)	78.7	74.0	76.2
LSTMs + GCNs (K=1), no gates	78.2	74.8	76.5
GCNs (no LSTMs), K=1	78.7	58.5	67.1
GCNs (no LSTMs), K=2	79.7	62.7	70.1
GCNs (no LSTMs), K=3	76.8	66.8	71.4
GCNs (no LSTMs), K=4	79.1	63.5	70.4

Table 2: SRL results without predicate disambiguation on the Chinese development set.

BiLSTMがあるなら1層で十分，ない場合たくさん必要

➡ BiLSTMですでに十分離れた関係が見えている？

実験結果

System	P	R	F_1
Lei et al. (2015) (local)	-	-	86.6
FitzGerald et al. (2015) (local)	-	-	86.7
Roth and Lapata (2016) (local)	88.1	85.3	86.7
Marcheggiani et al. (2017) (local)	88.6	86.7	87.6
Ours (local)	**89.1**	**86.8**	**88.0**
Björkelund et al. (2010) (global)	88.6	85.2	86.9
FitzGerald et al. (2015) (global)	-	-	87.3
Foland and Martin (2015) (global)	-	-	86.0
Swayamdipta et al. (2016) (global)	-	-	85.0
Roth and Lapata (2016) (global)	90.0	85.5	87.7
FitzGerald et al. (2015) (ensemble)	-	-	87.7
Roth and Lapata (2016) (ensemble)	90.3	85.7	87.9
Ours (ensemble 3x)	**90.5**	**87.7**	**89.1**

Table 3: Results on the test set for English.

アンサンブルする前からSotAなテスト性能

構造中の単語間の距離

6単語以上離れた述語項構造

➡ 英語20%/中国語30%

GCNによって”テレポート”できると

考えると9%/13%になる

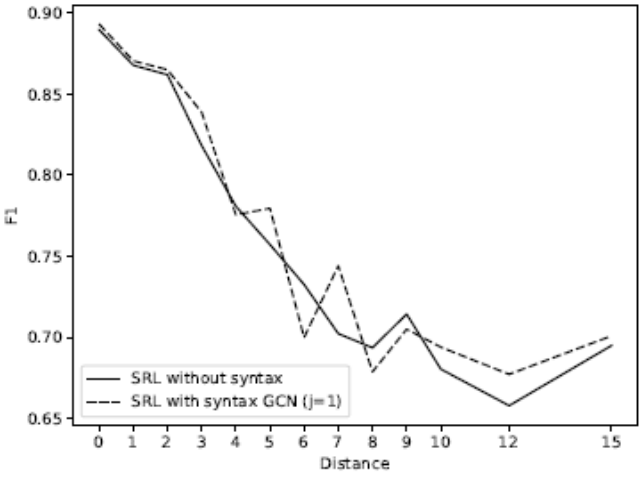


Figure 4: F_1 as function of word distance. The distance starts from zero, since nominal predicates can be arguments of themselves.

3層のBiLSTMによってすでに十分遠くが見えているせいで

あまりGCNを積み重ねても意味ないと考察している

実験結果

System	P	R	F_1
Lei et al. (2015) (local)	-	-	75.6
FitzGerald et al. (2015) (local)	-	-	75.2
Roth and Lapata (2016) (local)	76.9	73.8	75.3
Marcheggiani et al. (2017) (local)	79.4	76.2	77.7
Ours (local)	**78.5**	**75.9**	**77.2**
Björkelund et al. (2010) (global)	77.9	73.6	75.7
FitzGerald et al. (2015) (global)	-	-	75.2
Foland and Martin (2015) (global)	-	-	75.9
Roth and Lapata (2016) (global)	78.6	73.8	76.1
FitzGerald et al. (2015) (ensemble)	-	-	75.5
Roth and Lapata (2016) (ensemble)	79.7	73.6	76.5
Ours (ensemble 3x)	**80.8**	**77.1**	**78.9**

Table 5: Results on the out-of-domain test set.

外部ドメイン対象だとGCNなしの方がよかった

まとめ

- GCNを有向グラフ向けに拡張
  - エッジ依存の重み行列，親依存のゲート
- CoNLL-2009の述語構造解析に適用し無事SotA
- （この）GCNはとても簡単なので今後よく使われる？

感想

- 結局GCNの強みをあまり実感できていない気がする
- 本当にこの式・ネットワークでいいのか？
- Recursive NNと比較実験をしてほしかった

SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

Thomas N. Kipf
University of Amsterdam
T.N.Kipf@uva.nl

Max Welling
University of Amsterdam
Canadian Institute for Advanced Research (CIFAR)
M.Welling@uva.nl

3.1 EXAMPLE

In the following, we consider a two-layer GCN for semi-supervised node classification on a graph with a symmetric adjacency matrix A (binary or weighted). We first calculate $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ in a pre-processing step. Our forward model then takes the simple form:

$$Z = f(X, A) = \mathrm{softmax}\Big(\hat{A}\ \mathrm{ReLU}\Big(\hat{A}XW^{(0)}\Big)\,W^{(1)}\Big)\ . \tag{9}$$

Neural Message Passing for Quantum Chemistry

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$