Expert と Imitator の混合ネットワークによる大規模半教師あり学習

清野舜♠,♢　鈴木潤♠,♢　乾健太郎♠,♢
♠東北大学　♢理化学研究所 AIP センター
{kiyono, jun.suzuki, inui}@ecei.tohoku.ac.jp

1 はじめに

半教師あり学習 (Semi-supervised Learning：SSL) は，ラベル付きデータに加えてラベル無しデータを用いることでモデルの汎化性能の向上を目指す学習の枠組みである．一般に，自然言語処理ではラベル付きデータの作成には高度なアノテーション作業が必要となることが多く，大規模なデータの構築のコストが大きい．一方，ラベル無しデータは実質的に生文の集合で，Web のクローリングを通してテラバイト級の大規模なデータを比較的安価に獲得できる[*1]．このことを背景に，SSL は活発に研究されており，特に近年は SSL の深層学習への適用が盛んである [1, 2, 13].

本研究も同様に SSL の深層学習への適用に取り組む．特に本研究では，**ラベル無しデータ増加による汎化性能向上**を目標とする．一般論として，教師あり学習および SSL において，ラベル付きデータを増やすと汎化性能は向上する．しかし，SSL においてラベル無しデータの増加による汎化性能の向上は必ずしも自明ではない．また，仮に汎化性能を向上できたとしても，大規模なラベル無しデータを全て活用するためには，データ量に対してスケールする（大規模なデータを現実的な時間で学習できる）方法論が必要となる．つまり，ラベル無しデータの量を増やすことで性能が向上し，また高いスケーラビリティを持つ SSL が必要である．

しかしながら，既存の SSL においてラベル無しデータの量による性能への影響はあまり議論されてこなかった．例えば，ラベル無しデータをモデルの新たな訓練データとして活用する手法 [1, 11] では，ラベル無しデータの量は固定されているため，データ量の増減による性能への影響は言及されていない．また，ラベル付きデータとラベル無しデータを同じネットワークで用いるため，2 種類のデータには同じ計算コストが必要となり，スケーラビリティの観点からも大規模なデータを扱うのは困難である．

本研究では，スケーラブルな SSL として，Expert と Imitator の混合ネットワーク（Mixture of Expert/Imitator Networks; MEIN) を提案する．図 1 に，MEIN の概観図を示す．MEIN は，Expert ネットワーク（EXN）と複数の Imitator ネットワーク（IMN）の組み合わせから構成される．IMN は EXN よりも軽量なネットワークとして構成する．ここで，ラベル無しデータは個々の IMN を訓練するために用いるため，データの量に対してスケールすると期待できる．実験では，文書分類のベンチマークデータを用いて，MEIN による性能向上とそのスケーラビリティを議論する．

2 文書分類タスク

■タスク定義　いま，入力文章 $\boldsymbol{X}$ を 1-hot ベクトルからなる長さ T の系列とする．ここで，$\boldsymbol{x}_t \in \{0,1\}^{|\mathcal{V}|}$ は，入力 $\boldsymbol{X}$ の

[*1] 例えば，http://commoncrawl.org/ などが有名である

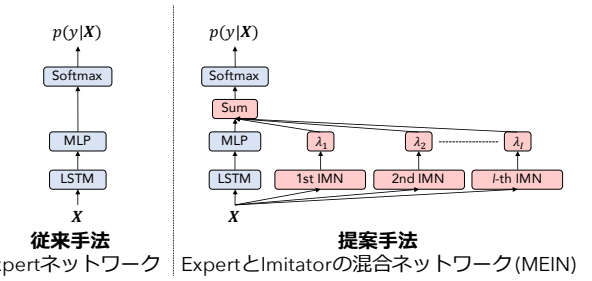


図 1: Expert と Imitator の混合ネットワーク（Mixture of Expert/Imitator Networks; MEIN）の概観図

t 番目のトークン（単語）を表す．また，$\mathcal{V}$ は語彙であり，$|\mathcal{V}|$ は語彙 $\mathcal{V}$ に含まれる単語数を表す．以降，$\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ の略記法として $(\boldsymbol{x}_t)_{t=1}^T$ を用いる．また，$\mathcal{Y}$ を出力クラスの集合として定義する．ここで，$y \in \{1, \ldots, |\mathcal{Y}|\}$ を出力クラスの番号を表すものとする．$\boldsymbol{X}_{a:b}$ を $\boldsymbol{X}$ の部分系列として定義する．つまり，$\boldsymbol{X}_{a:b} = (\boldsymbol{x}_a, \boldsymbol{x}_{a+1} \ldots, \boldsymbol{x}_b)$ かつ $1 \leq a \leq b \leq T$ である．また，$\boldsymbol{x}[i]$ を $\boldsymbol{x}$ 中の i 番目の要素として定義する．

■ベースラインモデル　ベースラインモデルとして，Miyato ら [11] と同様に LSTM ベースの手法を用いる．同手法は LSTM と多層パーセプトロン（MLP）から構成される．

まず，LSTM は入力 $(\boldsymbol{x}_t)_{t=1}^T$ を受け取り，隠れ層の系列 $(\boldsymbol{h}_t)_{t=1}^T$ を計算する．ここで，任意の t について $\boldsymbol{h}_t \in \mathbb{R}^H$ である．各時刻の演算は $\boldsymbol{h}_t = \mathrm{LSTM}(\boldsymbol{E}\boldsymbol{x}_t, \boldsymbol{h}_{t-1})$ であり，$\boldsymbol{E} \in \mathbb{R}^{D \times |\mathcal{V}|}$ は単語埋め込み行列である．また，D は単語埋め込み行列の次元数を表す．

次に，時刻 T の隠れ層 $\boldsymbol{h}_T$ が MLP に入力され，最終隠れ層 $\boldsymbol{s} \in \mathbb{R}^M$ が得られる．ここで，$\boldsymbol{s} = \mathrm{ReLU}(\boldsymbol{W}_h \boldsymbol{h}_T + \boldsymbol{b}_h)$ として計算される．$\boldsymbol{W}_h \in \mathbb{R}^{M \times H}$ はパラメータ行列，$\boldsymbol{b}_h \in \mathbb{R}^M$ はバイアス項である．M は最終隠れ層の次元数を表す．

最後に，条件付き確率を以下のように計算する．

$$z_y = \boldsymbol{w}_y^\top \boldsymbol{s} + b_y \tag{1}$$

$$p(y|\boldsymbol{X}, \boldsymbol{\Theta}) = \frac{\exp(z_y)}{\sum_{y' \in \mathcal{Y}} \exp(z_{y'})} \tag{2}$$

ここで，$\boldsymbol{w}_y \in \mathbb{R}^M$ はクラス y に対応する重み行列であり，b_y はバイアス項（スカラ値）である．また，$\boldsymbol{\Theta}$ は訓練対象のパラメータ集合を表す．

ベースラインモデルの訓練過程では，ラベル付きデータ集合 $\mathcal{D}_s$ の負の対数尤度を最小化するパラメータの探索を行う．この過程は，以下の最適化問題として記述することができる．

$$\boldsymbol{\Theta}' = \arg\min_{\boldsymbol{\Theta}} \left\{ L_s(\boldsymbol{\Theta}|\mathcal{D}_s) \right\} \tag{3}$$

$$L_s(\boldsymbol{\Theta}|\mathcal{D}_s) = -\frac{1}{|\mathcal{D}_s|} \sum_{(\boldsymbol{X}, y) \in \mathcal{D}_s} \log \left(p(y|\boldsymbol{X}, \boldsymbol{\Theta}) \right) \tag{4}$$

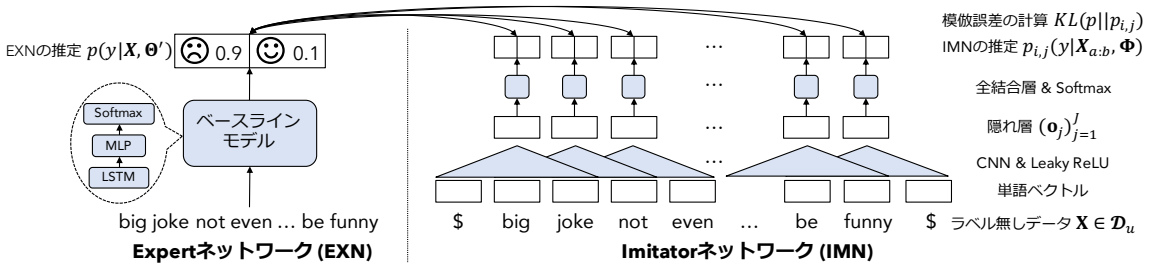


図 2: **1 番目の IMN（$c_1 = 1$）の概観図**: IMN は，制限された入力から，EXN の推定した確率分布の予測を行う．

ここで，$\boldsymbol{\Theta}'$ は最適化問題を解く過程で得られたパラメータ集合を表す．

3 提案手法

図 1 に提案手法である MEIN の概観を示す．MEIN は Expert ネットワーク（EXN）と複数の Imitator ネットワーク（IMN）から構成される．

■基本的なアイデア 一般に，ラベル付きデータはラベル無しデータと比較して少量しか手に入らない．そのため，ラベル付きデータだけで訓練した EXN では次のような問題が生じやすい；(a) 低頻度の特殊な特徴は，たとえ分類に寄与する特徴であっても少量のラベル付き訓練データから学習するのは困難である．(b) 少量のラベル付きデータでは、本来分類に寄与しない特徴（例: “this is a”）が偶然特定のラベルに偏って出現する可能性がある。ラベル付き訓練事例のみからの学習ではそうした「偽の」特徴を見抜くことが難しい．

MEIN では，EXN に加えて複数個の IMN を用意し，ラベル無しデータに対する IMN の予測が EXN の予測に近づくように IMN を訓練することによって，EXN の問題 (a) と (b) を同時に解決することを考える．IMN の核となるアイデアは次の 2 つである．第 1 に，個々の IMN は入力のごく一部の特徴（例えば，入力の部分文字列）だけからラベルを予測するように設計する．第 2 に，IMN の教師信号には，EXN が予測する離散的なラベルは用いず，EXN が予測するラベル事後確率を用いる．ここで，入力からの特徴抽出には，一貫したルールを適用すれば良いが，本研究では一例として，固定窓幅から入力を n-gram に分割することを考える．

IMN の訓練の概観を図 2 に示す．この図では，入力テキスト中の各 3-gram から EXN の出力を模倣するように IMN を訓練することを想定している．まず，EXN がラベル無し事例に対してラベルの事後確率分布を予測する（図 2 左）．次に，IMN が 3-gram から予測を行う．このとき，IMN が予測する事後確率が EXN の予測に近づくように IMN を訓練する（図 2 右）．ここで，ラベル付きデータで訓練された EXN は，上述 (a), (b) のような問題はあるとしても，少なくともラベル付き訓練事例に近い入力についてはある程度正確な予測ができる．そのため，(a) のような低頻度の有用な 3-gram は，EXN の模倣によって，ラベル無しデータから学習できると考えられる．また，(b) のような分類に寄与しない部分列に関しては，大量のラベル無しデータ上では特定のラベルへの偶然の偏りは起こりにくいと考えられる．そのため，分類に寄与しない「偽の」3-gram も IMN は見抜ける可能性がある．こうして訓練される IMN は、前述の問題 (a)、(b) を解消した特徴選択の役割を果たす．そこで最後に，IMN の訓練後，EXN と IMN の混合ネットワークの訓練を行うことで，各素性（IMN の出力）の持つ重みを調整する．

■ネットワーク構造 いま，$\sigma(\cdot)$ をシグモイド関数として次のように定義する：$\sigma(\lambda) = (1 + \exp(-\lambda))^{-1}$．$\boldsymbol{\Phi}$ は IMN の訓練可能なパラメータ集合であり，I は IMN の総数を表す．ここで，MEIN は次の条件付き確率をモデル化する．

$$p(y|\boldsymbol{X},\boldsymbol{\Theta},\boldsymbol{\Phi},\boldsymbol{\Lambda}) = \frac{\exp(z'_y)}{\sum_{y' \in \mathcal{Y}} \exp(z'_{y'})} \tag{5}$$

$$\text{where} \quad z'_y = z_y + \sum_{i=1}^{I} \sigma(\lambda_i)\boldsymbol{\alpha}_i[y] \tag{6}$$

λ_i は i 番目の IMN に対応するロジット $\boldsymbol{\alpha}_i$ の重みを制御するパラメータである．また，$\boldsymbol{\Lambda}$ を $\boldsymbol{\Lambda} = \{\lambda_1, \ldots, \lambda_I\}$ として定義する．ここで，ロジット $\boldsymbol{\alpha}_i$ はラベルの予測確率分布であり，これが素性として動作すると仮定する．ここで，式 6 はの第 1 項は，ベースラインモデルのロジット $z_y = \boldsymbol{w}_y^\top \boldsymbol{s} + b_y$（式 1）である．また，全ての i について $\sigma(\lambda_i) = 0$ とすることで，式 5 は $\boldsymbol{\Phi}$ の値によらず，式 2 と同値となる．

いま，c_i は i 番目の IMN の窓幅を表すとすると，i 番目の IMN は，入力 $\boldsymbol{X}$ から J 個の入力を幅 c_i の窓によって生成する．各 J 個の入力に対し，IMN は EXN の出力を予測し，合計 J 個の予測が得られる．また，i 番目の IMN のロジット $\boldsymbol{\alpha}_i$ は，これらの予測の平均によって計算する．具体的には，ロジット $\boldsymbol{\alpha}_i$ の定義は以下の通りである．

$$\boldsymbol{\alpha}_i = \log\left(\frac{1}{J}\sum_{j=1}^{J} p_{i,j}(y|\boldsymbol{X}_{a:b},\boldsymbol{\Phi})\right), \tag{7}$$
$$\text{where} \quad a = j - c_i \quad \text{and} \quad b = j + c_i$$

ここで，a と b はそれぞれ固定窓の始まりと終わりを表すスカラ値のインデックスである．

■IMN の定義 式 7 のモデル化には，任意のネットワーク構造を採用可能であり，IMN が EXN よりも軽量なネットワークとなればよい．本研究では，IMN の一例として単層 CNN ベースのネットワークを用いて $p_{i,j}(y|\boldsymbol{X}_{a:b},\boldsymbol{\Phi})$ をモデル化した．CNN は並列化が容易であり，高速な演算が可能である．

図 2 に IMN の概観を示す．まず，IMN は入力 $\boldsymbol{X}$ を単語埋め込みベクトルの系列として受け取り，隠れ層の系列 $(\boldsymbol{o}_j)_{j=1}^{J}$ を計算する．このとき，1 次元畳み込み [5] と活性化関数 Leaky ReLU を用いる．ここで，J を常に T と一致させるため，入力 $\boldsymbol{X}$ の先頭と終端を零ベクトル $\mathbf{0} \in \mathbb{R}^{|\mathcal{V}'| \times c_i}$ で補填する．$|\mathcal{V}'|$ は IMN の語彙に含まれる単語の数を表す．

各 IMN は入力を n-gram に分割するための固定窓幅 c_i を持つ．ここでは，i 番目の IMN に関して，任意の窓幅を設定できるが，本研究では，簡単のため $c_i = i$ とした．例えば，図 2 に示したように，IMN($i = 1$) は窓幅 $c_1 = 1$ を持つ．

次に，i 番目の IMN は，各隠れ層 $\boldsymbol{o}_j$ から $p_{i,j}(y|\boldsymbol{X},\boldsymbol{\Phi})$ から以下のように推定する．

$$p_{i,j}(y|\boldsymbol{X}_{a:b},\boldsymbol{\Phi}) = \frac{\exp(\boldsymbol{w}'^{\top}_{i,y}\boldsymbol{o}_j + b'_{i,y})}{\sum_{y'\in\mathcal{Y}}\exp(\boldsymbol{w}'^{\top}_{i,y'}\boldsymbol{o}_j + b'_{i,y'})} \quad (8)$$

ここで，$\boldsymbol{w}'_{i,y} \in \mathbb{R}^N$ は i 番目の IMN に対応する重みベクトルであり，スカラ値 $b'_{i,y}$ はクラス y のバイアス項である．また，N は CNN のカーネルの次元数を表す．

■訓練の枠組み まず，各 IMN の模倣誤差を，ラベル無しデータ $\boldsymbol{X}$ における EXN と IMN の予測確率分布間の KL ダイバージェンス $\mathrm{KL}(p(y|\boldsymbol{X},\boldsymbol{\Theta})||p_{i,j}(y|\boldsymbol{X}_{a:b},\boldsymbol{\Phi}))$ とする．ここで，模倣誤差が入力の各部分列 $\boldsymbol{X}_{a:b}$ に関して定義されていることに注意すると，制限された入力から EXN を模倣する，というアイデアが確かに実現されていることがわかる．

訓練の目的は，式 5 の負の対数尤度と模倣誤差をともに最小化するパラメータの推定である．したがって，以下の 2 つの最適化問題を同時に解く必要がある．

$$\widehat{\boldsymbol{\Theta}},\widehat{\boldsymbol{\Lambda}} = \arg\min_{\boldsymbol{\Theta},\boldsymbol{\Lambda}}\{L'_s(\boldsymbol{\Theta},\boldsymbol{\Lambda}|\widehat{\boldsymbol{\Phi}},\mathcal{D}_s)\} \quad (9)$$

$$\widehat{\boldsymbol{\Phi}} = \arg\min_{\boldsymbol{\Phi}}\{L_u(\boldsymbol{\Phi}|\boldsymbol{\Theta}',\mathcal{D}_u)\} \quad (10)$$

式 9 と 10 が示す通り，ラベル有り・無しデータによって別々のパラメータを更新する．具体的には，ラベル付きデータ $(\boldsymbol{X},y) \in \mathcal{D}_s$ を用いて EXN のパラメータ $\boldsymbol{\Theta}$ と IMN の係数 $\boldsymbol{\Lambda}$ を更新する．また，ラベル無しデータ $\boldsymbol{X} \in \mathcal{D}_u$ を用いて，IMN のパラメータ $\boldsymbol{\Phi}$ を更新する．

訓練の過程は以下に述べる 3 ステップからなる．まず，通常の教師あり学習の枠組みを用いて，ラベル付きデータから $\boldsymbol{\Theta}'$ を推定する．このとき，全ての i について $\lambda_i = -\infty$ とすることで，式 6 において $\sigma(\lambda_i) = 0$ とする．この過程は，ベースラインモデルのパラメータ推定（式 4）に相当する．次に，IMN 集合のパラメータ $\boldsymbol{\Phi}$ を式 10 の最適化問題を解くことによって推定する．具体的には，損失関数は以下の通りである．

$$L_u(\boldsymbol{\Phi}|\boldsymbol{\Theta}',\mathcal{D}_u) = \frac{1}{|\mathcal{D}_u|}\sum_{\boldsymbol{X}\in\mathcal{D}_u}\sum_{i=1}^{I}\sum_{j=1}^{J}\mathrm{KL}(p||p_{i,j}) \quad (11)$$

$$\mathrm{KL}(p||p_{i,j}) = -\sum_{y\in\mathcal{Y}} p(y|\boldsymbol{X},\boldsymbol{\Theta}')\log\left(p_{i,j}(y|\boldsymbol{X}_{a:b},\boldsymbol{\Phi})\right) + const \quad (12)$$

ここで，$\mathrm{KL}(p||p_{i,j})$ は模倣誤差の略記である．また，$const$ は $\boldsymbol{\Phi}$ に依存しない定数項である．

最後に，式 9 を解いて $\boldsymbol{\Theta}$ と $\boldsymbol{\Lambda}$ を推定する．損失関数は以下の通りである．

$$L'_s(\boldsymbol{\Theta},\boldsymbol{\Lambda}|\widehat{\boldsymbol{\Phi}},\mathcal{D}_s) = -\frac{1}{|\mathcal{D}_s|}\sum_{(\boldsymbol{X},y)\in\mathcal{D}_s}\log\left(p(y|\boldsymbol{X},\boldsymbol{\Theta},\widehat{\boldsymbol{\Phi}},\boldsymbol{\Lambda})\right) \quad (13)$$

4 実験

■データセット MEIN の効果を検証するため，(1) 評判分類タスク（SEC）(2) カテゴリ分類タスク（CAC）の 2 つを用いて実験を行った．評判分類タスクでは，ベンチマークデータとして IMDB[9]，Elec[4] と Rotten Tomatoes (Rotten)[12] を用いた．また，カテゴリ分類タスクでは RCV1[8] を用いた．各データセットにはラベル無しデータが付属している．表 1 に，各データセットの詳細をまとめた．

表 1: **データセットの詳細**: 各数値は，データセットに含まれるインスタンスの数を表す．

タスク	データセット	クラス数	訓練	開発	テスト	ラベル無し
SEC	Elec	2	22,500	2,500	25,000	200,000
	IMDB	2	21,246	3,754	25,000	50,000
	Rotten	2	8,636	960	1,066	7,911,684
CAC	RCV1	55	14,007	1,557	49,838	668,640

表 2: **実験結果**: 各数値はエラー率（%）を表す．ラベル無しデータを用いた手法は † で表記している．* が併記された値は，ADV-LM-LSTM と比較して統計的有意差があることを表す．Miyato と Sato はそれぞれ Miyato ら [11]，Sato ら [14] の論文値を引用した．

手法	Elec	IMDB	Rotten	RCV1
LSTM	10.09	10.98	26.47	14.14
LSTM+IMN†	**8.83**	**10.04**	**24.93**	**12.31**
LM-LSTM†	5.72	7.25	16.80	8.37
LM-LSTM+IMN†	**5.48**	**6.51**	**15.91**	**7.53**
ADV-LM-LSTM†	5.38	6.58	15.73	7.89
ADV-LM-LSTM+IMN†	**5.14***	**6.07***	**13.98**	**7.51***
VAT-LM-LSTM (再実装) †	5.47	6.20	18.50	8.44
VAT-LM-LSTM (Miyato)†	5.54	5.91	19.1	7.05
VAT-LM-LSTM (Sato)†	5.66	5.69	14.26	11.80
iVAT-LSTM (Sato)†	5.18	5.66	14.12	11.68

■ベースラインモデル（EXN） 以下に述べる 3 種類の EXN を用意し，それぞれ IMN と組み合わせた際の性能を評価した．

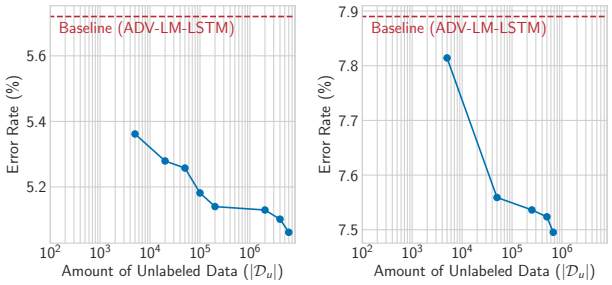
- LSTM：第 2 節で述べたベースラインモデルである．
- LM-LSTM：Dai ら [2] に従い，LSTM と単語埋め込み行列を訓練済みの RNN 言語モデル（LM）で初期化したモデルである．LM は，各データセットの訓練データとラベル無しデータによって訓練した．Miyato らと Sato ら [11, 14] は，同モデルをベースラインとして用いた．
- ADV-LM-LSTM：敵対性学習（Adversarial Training: ADV）[3] は，入力に微小な摂動を加えることで，ネットワーク全体をノイズに対して頑健にする効果がある．ADV は Miyato ら [11] によって文書分類に適用された．本研究では同モデルの再実装を用いる．

■ハイパーパラメータ 文献 [2, 11, 14] などで一般的に用いられている値を採用した．詳細は文献 [6] の表 2 を参照せよ．

EXN と IMN にはそれぞれ別の語彙を用いた．EXN の語彙 $\mathcal{V}$ は，既存研究に従い，訓練データ中で頻度 1 の単語を取り除くことで構築した [2, 11, 14]．また，IMN の語彙 $\mathcal{V}'$ は Byte Pair Encoding（BPE）[15] を用いて構築した[*2]．BPE のマージ過程は，各データセットのラベル付きデータとラベル無しデータから学習し，マージ回数は 20,000 回に設定した．

■実験結果 表 2 に実験結果を示す．評価指標としてはエラー率を用いたため，低い値ほど性能が良いことを表す．ここで，各値は乱数の種を変えて実験した際の 5 回平均である．評価時のパラメータとして，開発データ上で最も性能が良いエポックのものを採用した．既存研究との比較用

[*2] BPE の実装として，sentencepiece [7] を用いた．



(a) Elec (b) RCV1

図 3: **ラベル無しデータの総量に対するエラー率（%）の変化**:x 軸は対数スケールである．水平の破線は EXN（ADV-LM-LSTM）単体の性能を表す．

に，VAT-LM-LSTM[11] と iVAT-LSTM[14] の論文値と VAT-LM-LSTM（再実装）の値も報告する．

表 2 から，IMN はベースラインモデル（EXN）の性能を全てのベンチマークデータについて一貫して改善したことが分かる．また，IMN による性能の上がり幅は，EXN の性能に依存しないことが分かる．例えば，表 2 より，Rotten データセット上で，IMN は LSTM，LM-LSTM と ADV-LM-LSTM の性能をそれぞれ 1.54%, 0.89%，1.22% 改善したと分かる．このことから，IMN はより強力な EXN の性能も改善できると期待される．また，ADV-LM-LSTM+IMN は，全てのデータセット上で VAT-LM-LSTM よりも高い性能を示しただけでなく，Elec と Rotten データセット上において，既存手法と比較して最高性能を達成した．

5 分析

■ラベル無しデータの量とモデルの性能 MEIN が**ラベル無しデータ増加による汎化性能向上**を実現しているかを調査するため，IMN の訓練に用いるラベル無しデータの総量を変化させた場合の性能を調査した．

分析には Elec と RCV1 データセットを用いた．Elec から {5K, 20K, 50K, 100K, データ全体 }，RCV1 から {5K, 50K, 250K, 500K, データ全体 } のラベル無しデータの集合をそれぞれサンプリングして作成した．また，Elec に関しては Amazon Review データセット [10] の Electronics セクションから追加でラベル無しデータをサンプリングし，{2M, 4M, 6M} のデータセットを作成した．各データに関し ADV-LM-LSTM+IMN の訓練と評価を行った．

図 3a と 3b より，ラベル無しデータの総量を増やすことで EXN の性能の上がり幅が増加することが分かる．特に図 3a 上，6M のラベル無しデータで訓練した ADV-LM-LSTM+IMN が 5.06% のエラー率を達成した．これは表 2 の最高性能 (5.14%) よりも良い値である．これらの結果から，MEIN により**ラベル無しデータ増加による汎化性能向上**が実現されていることが分かる．

■IMN の計算速度 IMN の計算速度を EXN と他の SSL 手法（VAT）と比較することで，提案手法のスケーラビリティに関する分析を行った．具体的には，訓練過程において各ネットワークが毎秒に処理するトークンの数を計測した．各計測では NVIDIA Tesla V100 GPU を用いた．

表 3 に計測結果を示す．表から，最も計算の遅い IMN ($c_i = 1, 2, 3, 4$) であっても，LSTM と比較して 1.8 倍高速に

表 3: **各モデルが処理するトークン数の比較（訓練時）**

手法	トークン数/秒	相対速度
LM-LSTM	41,914	-
ADV-LM-LSTM	13,791	0.33x
VAT-LM-LSTM	9,602	0.23x
IMN ($c_i = 1$)	555,613	13.26x
IMN ($c_i = 1, 2$)	236,065	5.63x
IMN ($c_i = 1, 2, 3$)	122,076	2.91x
IMN ($c_i = 1, 2, 3, 4$)	75,393	1.80x

動作すること分かる．また，同モデルは VAT-LM-LSTM と比較すると 8 倍高速である．この結果から，大規模なラベル無しデータに対しても，現実的な時間での訓練が可能であることが期待される．また，各 IMN は並列に訓練可能である．そのため，複数枚の GPU を用いることで，IMN を表 3 よりもさらに高速に訓練することができる．

6 おわりに

本研究では，新しい半教師あり学習の手法を提案した．提案手法（MEIN）は，ベースラインモデル（EXN）と，複数の補助的なネットワーク（IMN）の組み合わせから構成する．文書分類のベンチマークデータセットを用いた実験で，提案手法が複数の EXN の性能を改善することを示した．また，ラベル無しデータを増やすことで汎化性能が向上することを確認した．提案手法は VAT と比較して 8 倍高速に動作するため，大規模なラベル無しデータに対してもスケールする枠組みである．今後はより大規模なラベル無しデータに対する提案手法の効果を検証したい．

参考文献

[1] Kevin Clark, Thang Luong, and Quoc V. Le. Cross-View Training for Semi-Supervised Learning. In *ICLR*, 2018.
[2] Andrew M Dai and Quoc V Le. Semi-supervised Sequence Learning. In *NIPS*, pp. 3079–3087, 2015.
[3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *ICLR*, 2015.
[4] Rie Johnson and Tong Zhang. Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding. In *NIPS*, pp. 919–927, 2015.
[5] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A Convolutional Neural Network for Modelling Sentences. In *ACL*, pp. 655–665, 2014.
[6] Shun Kiyono, Jun Suzuki, and Kentaro Inui. Mixture of expert/imitator networks: Scalable semi-supervised learning framework. *CoRR*, Vol. abs/1810.05788, , 2018.
[7] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *EMNLP*, pp. 66–71, 2018.
[8] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A New Benchmark Collection for Text Categorization Research. *JMLR*, Vol. 5, pp. 361–397, 2004.
[9] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In *ACL*, pp. 142–150, 2011.
[10] Julian McAuley and Jure Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *RecSys*, pp. 165–172, 2013.
[11] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial Training Methods For Semi-Supervised Text Classification. In *ICLR*, 2017.
[12] Bo Pang and Lillian Lee. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *ACL*, pp. 115–124, 2005.
[13] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *NAACL*, pp. 2227–2237, 2018.
[14] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. Interpretable Adversarial Perturbation in Input Embedding Space for Text. In *IJCAI*, pp. 4323–4330, 2018.
[15] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *ACL*, pp. 1715–1725, 2016.