Human-in-the-Loop ニューラルトピックモデリング

米谷　亜夕†　　伊藤　寛祥††　　森嶋　厚行††

† 筑波大学 情報学群　〒 305-8550 茨城県つくば市春日 1-2
†† 筑波大学 図書館情報メディア系　〒 305-8550 茨城県つくば市春日 1-2
E-mail: †s2213614@u.tsukuba.ac.jp, ††{ito,mori}@slis.tsukuba.ac.jp

あらまし　トピックモデルは文書集合に含まれる潜在的なトピックを抽出するモデルであり，テキストマイニングの手法の一つとして知られている．しかしながらトピックモデルで抽出されるトピックは， 人間にとって必ずしも好ましい分析結果になるとは限らない．本研究では，ニューラルトピックモデルに対して， Human-in-the-Loop によって人間からのフィードバックをモデルに反映する手法を提案する．本論文では特に，トピックへの単語の追加・削除について検討する．ニューラルトピックモデルに対して Human-in-the-Loop を適用することで，勾配法に基づくモデルの最適化が可能になり，モデルの拡張が容易になるという利点が存在する．実験では，ベースラインとするトピックモデルと本研究で提案する Human-in-the-Loop ニューラルトピックモデルを Perplexity と人間のフィードバックの反映率を評価指標として用いて比較実験する．結果として，ベースラインとした LDA と比較して提案手法はより良い Perplexity を獲得し，反映率は 0.9 以上のスコアを獲得することができた．

キーワード　トピックモデル，Human-in-the-Loop，テキストマイニング, 情報抽出

1　はじめに

トピックモデルは潜在的なトピックを抽出するモデルでテキストマイニング手法の一つとして知られていおり，様々な領域に応用されている．例えば，文書分類 [1] やテキスト分析 [2]，推薦システム [3] のような分野では必要不可欠な技術である．トピックモデルはこれまでいくつか提案されてきた．潜在意味解析（Latent Semantic Indexing: LSI）[4] や確率的潜在意味解析（Probabilistic Latent Semantic Indexing: PLSI）[5]，潜在ディリクレ配分法（Latent Dirichlet Allocation: LDA）[6] は特に代表的なトピックモデルである．

ここでは，本研究と関連の深いトピックモデルである LDA について紹介する．LDA は，文書の確率的生成モデルとして提案された，よく用いられるトピックモデルの 1 つである．LDA では，各文書中に潜在的なトピックがあると仮定し，統計的に共起しやすい単語集合を潜在的なトピックとする確率変数を用いて表現する．これにより，データセット中に存在する潜在的なトピックを抽出でき，データセット全体を要約する表現が得られる．また，LDA は文書の形式に限らず，共起情報があるようなデータ全般に適応することができるモデルである．そのため，自然言語処理のみならず，画像処理や音声処理，情報検索などのさまざまな分野で応用されている．

LDA に対して，ニューラルトピックモデル（Neural Topic Model: NTM）[7] [8] [9] は，変分オートエンコーダ（Variational Auto Encoder: VAE）[10] によって潜在的なトピックを推定することができるモデルである．VAE は変分ベイズ推論とニューラルネットワークを組み合わせたモデルであり，深層学習による生成モデル 1 つである．VAE によってより複雑な推論を行うことが可能な上，確率的勾配法でモデルの最適化が行えることやニューラルなモデルとの組み合わせなどの拡張が容易である．しかし， LDA や NTM を用いたトピックの抽出においては，人間にとって必ずしも好ましい分析結果になるとは限らない．

先行研究において，トピックモデルに Human-in-the-Loop を組み込んだ手法である Human-in-the-Loop トピックモデル（Human-in-the-Loop Topic Model: HL-TM）が注目を集めている [11] [12] [13]．Human-in-the-Loop とは，人間を何らかのシステムに組み込むこのによって，人間にとってより好ましい結果に近づける方法である．トピックモデルに Human-in-the-Loop を取り入れる場合，抽出されたトピックが人間にとって好ましくないものにならないように，トピックに単語を追加・削除する，といった操作を人間が加えることによってシステムを補助する．既存研究では，確率的生成モデルに基づくトピックモデルをベースにしたものが提案されているが， NTM に基づく Human-in-the-Loop トピックモデルは提案されていない．

そこで，本研究では NTM の学習のプロセスに Human-in-the-Loop 組み込んだ Human-in-the-Loop ニューラルトピックモデリング（Human-in-the-Loop Neural Topic Modeling: HL-NTM）を提案する．本研究では特に，Human-in-the-Loop によって，人間が必要と考える単語をトピックに追加する操作と，人間が不必要と考える単語をトピックから削除する方法について検討する．HL-NTM は人間のフィードバックに対する重みのパラメータをもち，このパラメータを調整することで，人間のフィードバックをどれだけ取り入れるかを調整可能である．さらに， NTM に基づく手法を構成することで，トピックモデルを用いたニューラルネットワークに基づく手法に対して本手法を適用することが可能になる．

実験では Livedoor ニュースコーパスを用い，トピックモデルの予測精度を示す指標である Perplexity と，モデルに反映

されたフィードバック数と反映されなかったフィードバック数に基づいて算出される反映率を用いて評価を行った．実験の結果， HL-NTM が既存の手法と比較してモデルの予測精度を維持しながら，優れた反映率のスコアを得られることが示された．

本研究の貢献をまとめると，以下の通りである．

（1） 本論文では NTM に Human-in-the-Loop 組み合わせた構造を持つ， Human-in-the-Loop ニューラルトピックモデリングを提案する．

（2） ベースラインとした LDA と比較して，より優れた Perplexity のスコアを保ちながら Human-in-the-Loop の構造を組み込むことが可能であることを，ランダムな操作を行うユーザを想定した実験によって示した．

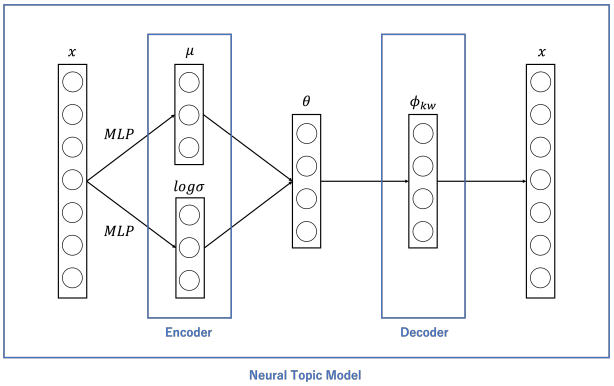

図 1: NTM のアーキテクチャ図．

2 関 連 研 究

本節では，トピックモデルに関する内容の中でも，特に本研究に関係が深い LDA と NTM について説明する．加えて，Human-in-the-Loop とトピックモデルを組みわせたモデルの提案をしている論文をいくつか挙げ，その内容について触れる．また，各項で紹介するトピックモデルに対して，本研究との関連性と新規性について改めて述べる．

2.1 トピックモデル

トピックモデルとは，自然言語処理の分野で用いられる潜在意味解析手法のひとつで文書集合のトピックを分析する手法である [4] [5] [6]．トピックモデルでは，文章を複数の単語の集合であると捉え，それらの単語の共起性に着目して文章をいくつかのクラスに分類する．トピックモデルは，文章が複数の潜在的なトピックから構成され，それらは確率的に生成されると仮定し，単語がそのトピック分布に従って出現するという構造になっている．

最も代表的なトピックモデルとしては，潜在ディリクレ配分法（LDA）[6] が知られている．LDA は，これまで拡張した手法がさまざま提案されてきている．例えば，トピック間の相関をモデル化した Correlated Topic Model（CTM）[14] や文書中の単語の順序も考慮できるように系列データを扱える隠れマルコフモデル（Hidden Markov Model: HMM）と LDA を組み合わせた HMM-LDA [15] などが挙げられる．

しかし， LDA のような統計的手法に基づくトピックモデルに新しく Human-in-the-Loop を組み合わせて推論をするためには，モデルをもう一度構築し直さなければならない．

ニューラルトピックモデル（NTM）は， VAE を元に構築されたトピックモデルである． LDA に対して NTM は，モデル構築の度に最適化アルゴリズムの導出を行う必要がなく，勾配法に基づいてモデルの最適化を行うことが可能である．また，ニューラルなモデルであり，他のニューラルな手法での拡張が容易なため， LDA の問題点を解決した． NTM に関しての詳細な説明は，3.1 項で述べる．

2.2 Human-in-the-Loop トピックモデル

Human-in-the-Loop トピックモデル（HL-TM）において，既存の研究で提案されている HL-TM についてここでは 3 つ紹介する．

HL-TM のシステム評価に関する研究として， Varun Kumar 氏らによって提案された HL-TM システム [11] では，ユーザの想定に近い出力になったかどうかを評価するための指標を導入している． Khan Muhammad Haseeb Ur Rehman 氏らによって提案されたキーフレーズによる文書分類を可能とした HL-TM [12] では，リファインメント操作が実際の応用上重要なドキュメント-トピックの関連にほとんど影響を与えないことを指摘し，ドキュメント-トピックの効果的な改善を意図したキーフレーズベースのリファインメント関数を提案している．また， Zheng Fang 氏らは，ユーザが全てのステップを比較・記録できるユーザフレンドリーなインターフェースと，ユーザがフィードバックを反映させるための新しいトピック提案機能を備えた，新しいインタラクティブな HL-TM を開発している [13]．

2.3 本研究の位置付け

これらの研究では，これまで提案されてきた統計的手法を用いたトピックモデルを元に Human-in-the-Loop を組み込んだトピックモデルとなっている．したがって，本研究の位置付けとしては， HL-NTM の基本的な構築とその有用性を示すことを目的とする．

3 事 前 知 識

本節では，前提知識となる NTM について詳しく説明する．NTM は本研究において， Human-in-the-Loop を組み込む上で重要な基礎知識である．

3.1 ニューラルトピックモデル

NTM は， 変分自己符号化器 （Variational Auto Encoder: VAE） を用いて潜在トピックを抽出するニューラルなトピックモデルである．VAE は，次のような仕組みでデータ x を生成する．まず，潜在変数 z が分布 $p(z)$ に従ってランダムに生成されるとする．この生成された z を条件付き分布 $p(x|z)$ に与えると，x が生成されるという仕組みである．また，条件付き分布 $q(z|x)$ を考え，条件付き分布 $p(z|x)$ との 2 つの分布の距離を

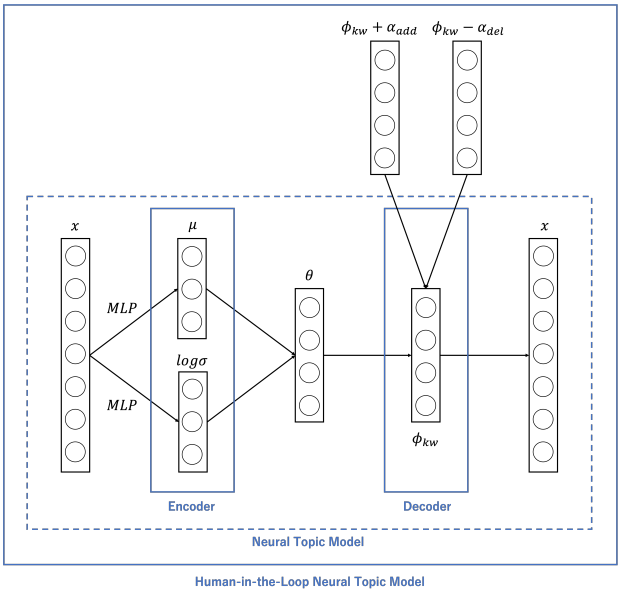

図 2: HL-NTM のアーキテクチャ図.

測る KL ダイバージェンスを考える．$q(z|x)$ はニューラルネットワークで表現する．損失関数は式（1）のように定義される．

$$\mathcal{L}_{NTM} = D_{KL}[q(z|x)||p(z)] - \mathbb{E}_{q(z|x)}[\log p(x|z)] \quad (1)$$

基本的に VAE と同じ形であり，KL ダイバージェンス損失とクロスエントロピー誤差の損失を組み合わせた関数である．条件付き分布 $q(z|x)$ から z を生成するプロセスはエンコーダ，条件付き分布 $p(x|z)$ から x を生成するプロセスはデコーダである．

ここで， NTM のアーキテクチャを図 1 に示す. アーキテクチャ図を元に， NTM の構造について詳しく説明する．文書集合のそれぞれの文書 x から BoW ベクトル $x_{bow} \in \mathbb{R}^V$ を生成し，これを入力とする．V は文書集合に含まれる単語の種類数（語彙数）である．NTM のエンコーダについて説明する．まず，$p(z|x)$ が標準正規分布 $q(z)$ に従うと仮定する．すると，$p(z|x)$ のパラメータ μ と $\log\sigma$ は，x_{bow} を入力として学習される．μ と $\log\sigma$ は式（2）である．

$$\mu = f_\mu(f_e(x_{bow})), \quad \log\sigma = f_\sigma(f_e(x_{bow})) \quad (2)$$

$f_*(\cdot)$ の全結合層は ReLU を活性化関数とする．

Decoder は，最終的に x_{bow} を再構成する．潜在変数 z は式（3）のようにサンプリングされる．

$$z \sim \mathcal{N}(\mu, \sigma^2) \quad (3)$$

トピック分布を式（4），x_{bow} の再構成は式（5）に示す．

$$\theta = softmax(f_\theta(z)) \quad (4)$$

$$x_{bow} \sim softmax(f_\phi(\theta)) \quad (5)$$

式（4）は潜在変数 z によって導出されるトピック分布である．（5）のパラメータ ϕ は θ を元に出力される，トピックごと単語分布となる．エンコーダと同様，$f_*(\cdot)$ の全結合層は ReLU を活性化関数とする．

4 提 案 手 法

本研究では，ニューラルトピックモデルに Human-in-the-Loop の構造を組み合わせた Human-in-the-Loop ニューラルトピックモデル（HL-NTM）を提案する．本節では， HL-NTM のアーキテクチャ， Human-in-the-Loop として導入する操作，モデルの学習方法について説明する．

4.1 HL-NTM のアーキテクチャ

本項では， HL-NTM のアーキテクチャについて説明する．HL-NTM のアーキテクチャは図 2 に示す．基本的には NTM と同じような構造をとる． NTM との違いは Human-in-the-Loop の機能として，デコーダ側に人間のフィードバックの反映に対する損失関数 $\phi_{kw}+\alpha_{add}$， $\phi_{kw}-\alpha_{del}$ が追加される点である．HL-NTM は事前学習済みの NTM をもとに，デコーダ側で人間のフィードバックを反映するようにファインチューニングを行う．デコーダ側に追加されるハイパーパラメータ等に関しては，4.2 項で解説する．

4.2 Human-in-the-Loop について

本研究では， HL-NTM における Human-in-the-Loop の操作として，ユーザによるトピックに対する単語の追加・削除を想定する．それぞれの操作についてより具体的に述べる．

（1） 単語の追加　　ユーザによって指定された，k 番目のトピックから削除する単語の集合を $\mathcal{W}_{add}^{(k)} = \{w_1, w_2, \cdots, w_n\}$ とする．

（2） 単語の削除　　ユーザによって指定された，k 番目のトピックに追加する単語の集合を $\mathcal{W}_{del}^{(k)} = \{w_1, w_2, \cdots, w_n\}$ とする．

4.3 学　　習

本項では， HL-NTM の学習方法について述べる．学習に用

いる損失関数は式（6）のように定義する．

$$\mathcal{L}_{HitL} = \mathcal{L}_{NTM} + \mathcal{L}_{add} + \mathcal{L}_{del} \tag{6}$$

ここで，$\mathcal{L}_{NTM}$ は NTM の損失関数である．また，$\mathcal{L}_{add}$ と $\mathcal{L}_{del}$ はそれぞれ単語の追加，単語の削除のための損失関数であり，式（7）,（8）のように定義される．

$$\mathcal{L}_{add} = \sum_{k=1}^{K} \sum_{w \in \mathcal{W}_{add}^{(k)}} \|\phi_{kw} + \alpha_{add}\| \tag{7}$$

$$\mathcal{L}_{del} = \sum_{k=1}^{K} \sum_{w \in \mathcal{W}_{del}^{(k)}} \|\phi_{kw} - \alpha_{del}\| \tag{8}$$

ここで，ϕ_{kw} は k 番目のトピックの w に対するデコーダのパラメータであり，α_{add}，α_{del} はハイパーパラメータである．ハイパーパラメータの数を大きくすればするほど，操作する単語に加わる重みが大きくなる．

よって，学習に用いる損失関数である式（6）は， NTM の損失，単語の追加を行った分の損失，単語の削除を行った時の損失を加算した式で構成されている．

5 実　　験

実験は，提案する HL-NTM と今回ベースラインとする LDA と NTM のスコアを比較して，モデルの予測精度を落とさずに Human-in-the-Loop の操作が反映可能か検証する．データセットには， livedoor ニュースコーパスを使用した．モデルの評価指標としては，予測精度を評価する Perplexity とモデルに反映されたフィードバック数と反映されなかったフィードバック数によって算出する反映率を用いることによって， HL-NTM の有用性を確かめる．

5.1 データセット

データセットは， livedoor ニュースコーパスを使用する．livedoor ニュースコーパスは， NHN Japan 株式会社が運営する livedoor ニュース記事を収集して作られ， 可能な限り HTML タグを取り除かれているものである． ニュース記事は，トピックニュース， Sports Watch， IT ライフハック， 家電チャンネル， MOVIE ENTER， 独女通信， エスマックス， livedoor HOMME， Peachy である． 収集時期はいずれも 2012 年 9 月上旬である．livedoor ニュースコーパスに対して janome Tokenizer を用いて， URL の除去， ストップワードの除去， 品詞の抽出の処理を行った． janome Tokenizer の処理によって除去した単語数は 928 語であり，今回学習に使用した総単語数は 20220 語である．

5.2 評 価 指 標

評価指標として，モデルの予測精度を測る Perplexity と，モデルに反映されたフィードバック数と反映されなかったフィードバック数によって算出する反映率を用いる．Perplexity は，式（9）のように定義される．

$$Perplexity = \exp\left\{-\frac{\sum_{d \in \mathcal{D}} \sum_{w \in d} \log p(w \mid \mathcal{M})}{\sum_{d \in \mathcal{D}} n_d}\right\} \tag{9}$$

ここで，$\mathcal{D}$ は文書集合，w は文書 d に含まれる単語，$p(w \mid \mathcal{M})$ は，モデル $\mathcal{M}$ による単語 w の生成確率である． Perplexity は小さいほど良い性能を示す指標となっている．ランダムな単語を返すモデルでは文書の語彙数，最小では 1 をとる． Perplexity が 1 の場合は，正しい単語の予測確率が 100% であるということになるため，単語は 1 択に絞られていると解釈することができる．

人間によるフィードバックのうち，出力されたトピック内の単語に対してフィードバックが反映された割合を 反映率（Reflection Rate）で評価する． 定義は（10）の通りである．

$$ReflectionRate = \frac{TP}{TP + FN} \tag{10}$$

ここで， TP はモデルに反映されたフィードバックの数， FN はモデルに反映されなかったフィードバックの数である．反映率は，実際の正解のうち，モデルによる予測がどれだけの割合で positive であったかを表現した指標である． 0.0（0%）-1.0（100%） の範囲で値をとり，大きいほど良い性能を示すことになる指標となっている．単語 1 つの操作を反映するフィードバックを例とすると，反映された場合は， $TP = 1$， $FN = 0$ であるため，反映率 $= 1$ となるため，単語が反映されたということがわかる．

5.3 実 験 設 定

実験比較には，提案手法の HL-NTM に対して，ベースラインとするトピックモデルは LDA と NTM である．これらのトピックモデルは前述の通り， Human-in-the-Loop を組み合わせていないシンプルなトピックモデルである．よって，Human-in-the-Loop の操作を組み込んでいないこれらのトピックモデルと比較して HL-NTM の Perplexity が悪化しないこと確認する．また， Perplexity を悪化させないように設定した状態のときのモデルに Human-in-the-Loop を組み込んだ場合のフィードバックの反映度合いを反映率によって示す．

各モデルに関する実験設定等について説明する．LDA はトピック抽出の実験を 10 回行う． LDA の Perplexity は，10 回の実験における平均値を結果とする． Perplexity はテストデータによって算出されたスコアである．また，今回の LDA による予測では，20 のトピックと各トピック内で出現する上位 10 単語が挙げられる．

NTM のパラメータに関して，隠れ層は 1000，トピック数は 20，トピック内で挙げられる単語数を 10 とする．学習に関するパラメータに関して，バッチサイズは 32，学習率は 0.001，エポック数は 150， spasity は 0.23 としている．それぞれ，学習のプロセスには，式（5）の f_ϕ の重みに対する L1 ペナルティが追加されている． L1 ペナルティを追加することによって，各トピック内の重みがゼロに近い単語は単語の情報量を削減し，学習の途中で f_ϕ の重みをスパースにしている．スパースにすることによって，重要度の低い単語や高い単語の重みをより極端な値にして，単語の情報を 2 分することができる．損失関数， Perplexity， L1 ペナルティに関しては，学習を行った結果のうち，平均的な学習のプロセスの出力結果を示す．

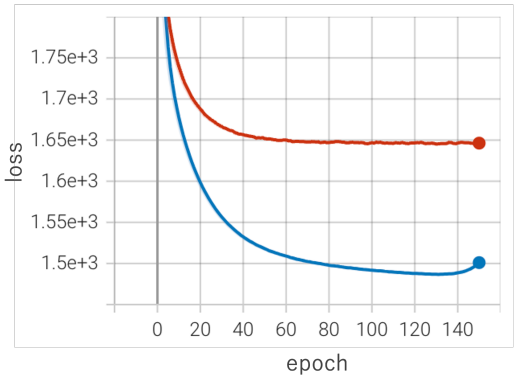

図 3: NTM による学習中の損失関数の推移．青は訓練データ，赤は検証データ．

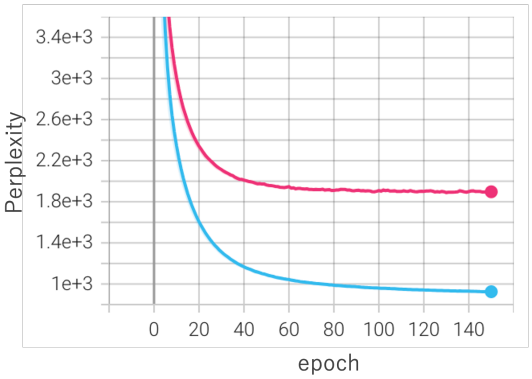

図 4: NTM による学習中の Perplexity の推移．青は訓練データ，赤は検証データ．

	Perplexity	
	LDA	NTM
平均	1841.9	1720.3

表 1: LDA と NTM の Perplexity 平均スコア．

NTM は，トピック抽出の実験を 10 回行う． NTM の Perplexity は，10 回の実験における平均値を結果とする． Perplexity はテストデータによって算出されたスコアである．また，今回の NTM による予測では，20 のトピックと各トピック内で出現する上位 10 単語が挙げられる．

HL-NTM は，事前学習を行なった NTM をベースに，デコーダの更新を行うように学習を行うことで Human-in-the-Loop の操作を導入する．事前学習を行なった NTM のパラメータは，前述したベースラインとする NTM の学習で設定したものとパラメータと同様に設定している．本実験では，単語の追加操作と単語の削除操作を追加する操作を Human-in-the-Loop とする．それぞれ，ランダムに選ばれたトピックにランダムに選ばれた単語を追加する操作とランダムに選ばれたトピックにランダムから選ばれた単語を削除する操作を分けて行う．ただし，ランダムに選出された単語がすでに選択されていた場合は単語の再選出を行う．単語の操作を行うのはそれぞれのトピックに出現する上位 10 単語に対してである．α_{add} のハイパーパラメータは 10，20，50，100，200 とし，α_{del} のハイパーパラメータは，0.1，0.2，0.5，1，3 とし，単語の操作数は 1，10，50，100，150 とする．トピックと単語の抽出の実験はそれぞれ 10 回行う． HL-NTM の Perplexity は，ハイパーパラメータ別に出力する． Perplexity はテストデータによって算出されたスコアである．反映率も同じようにして，ハイパーパラメータと単語の操作数を設定してハイパーパラメータ別に出力を行う．

5.4 実験結果

本項では， LDA， NTM， HL-NTM の各トピックモデルで行った実験結果を示し，その結果の特徴について述べる．

LDA， NTM で行なった 10 回の実験で出力された Perplexity を表 1 に示す． Perplexity は小数点第 2 位を四捨五入している．

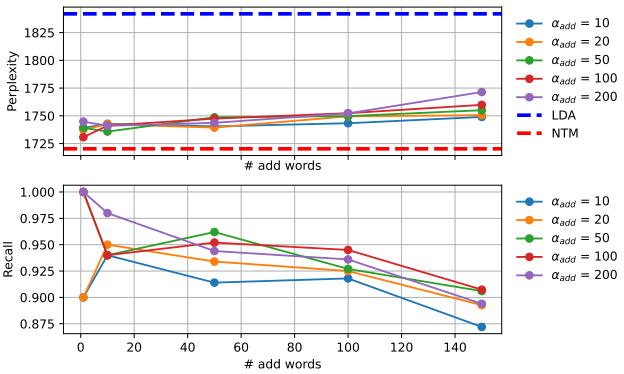

図 5: HL-NTM で単語の追加操作を行った時の Perplexity と反映率ののスコア．

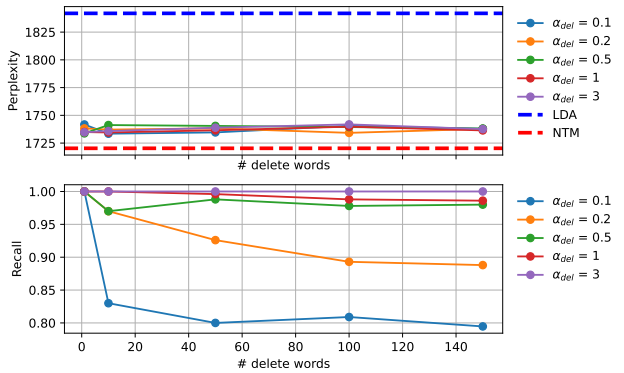

図 6: HL-NTM で単語の削除操作を行った時の Perplexity と反映率ののスコア．

NTM の学習プロセスの出力結果を，図 3，4 に示す．それぞれ，青は訓練データ，赤は検証データを入力とする学習のプロセスである． NTM の学習過程の loss の値を示す図 3 では，150 エポック付近になると損失関数の収束の傾向が見られる．図 4 も学習が進むにつれて同じように収束している．

HL-NTM の実験結果について，単語の追加操作と単語の削除操作を行った結果をそれぞれ示す．HL-NTM で単語の追加操作を行った実験結果として出力された Perplexity と反映率のスコアを図 5 に示す．それぞれ縦軸が Perplexity のスコアと反映率のスコア，横軸は単語の追加操作数である．今回設定し

たハイパーパラメータ数は 5 つのため，5 パターンの結果を出力している． Perplexity に関して， ベースラインとする LDA と NTM の Perplexity と HL-NTM の Perplexity を比較するために，グラフ上にそれぞれ破線で示している．

単語の追加操作を行った場合の Perplexity について，単語の操作数を増やすほどスコアが悪化する傾向が見られた．しかし，ハイパーパラメータの違いによってスコアの悪化度合いが大きく変化することは無かった．3 つのトピックモデルの Perplexity を比較すると， LDA の Perplexity が最も大きくなった． NTM の Perplexity が最も小さいため，最も良いスコアであるということがわかる．また， NTM と HL-NTM の Perplexity のスコアは近い値となり， HL-NTM はわずかに高くなった．反映率について，単語の操作数を増やすほどスコアが悪化する傾向が見られた．ハイパーパラメータを大きくするほど，反映率のスコアは 1 に近づき，加えた操作が出力された単語の結果に反映されていることがわかる．

HL-NTM で単語の削除操作を行った実験結果として出力された Perplexity と反映率のスコアを図 6 に示す． それぞれ縦軸が Perplexity のスコアと反映率のスコア， 横軸は単語の削除操作数である．単語の追加操作と同じように，今回設定したハイパーパラメータ数は 5 つのため，5 パターンの結果を出力している． Perplexity に関しても，ベースラインとする LDA と NTM の Perplexity と HL-NTM の Perplexity を比較するために，グラフ上にそれぞれ破線で示している．

単語の削除操作を行った場合の Perplexity について，単語の操作数を増やしてもスコアが悪化する傾向は見られなかった．加えて，ハイパーパラメータの違いによってもスコアの悪化度合いが大きく変化することは無かった．3 つのトピックモデルの Perplexity を比較すると， LDA の Perplexity が最も大きくなった． NTM の Perplexity が最も小さいため，良いスコアであるということがわかる．また， NTM と HL-NTM の Perplexity のスコアは近い値となり、こちらも単語の追加操作と同様に HL-NTM の Perplexity はわずかに高くなった．反映率について，単語の操作数を増やすほどスコアが悪化する傾向が見られた．ハイパーパラメータを大きくするほど，反映率のスコアは 1 に近づき，加えた操作が出力された単語の結果に反映されていることがわかる．

5.5 考　　察

本項では，5.4 項で述べた実験結果について考察する．

LDA の Perplexity と比較した場合， HL-NTM の Perplexity はスコアが大幅に下回った．よって， HL-NTM はモデルの予測精度を落とさずにフィードバック操作を反映させることができた． NTM と HL-NTM を比較すると， HL-NTM の Perplexity は NTM の Perplexity を僅かに上回る結果となった．このような結果になった理由としては， 事前学習済みの NTM からランダムに操作を加えて出力される単語に変更を加えたため，結果としては予測精度を落とすような操作を加えているということになるためであると考えられる．また，HL-NTM の Perplexity は NTM のスコアを大きく上回ってはいないため，予測精度をある程度保ちながら HL-NTM を構築することができたと考えることができる．

単語の追加操作の Perplexity は単語の削除操作の Perplexity と比較して， 単語の操作数が増えると悪化の傾向が見られた．今回は，ランダムなトピックとランダムな単語を選出して追加操作を行っているため，トピックに関係のない単語が追加されている可能性が高く，悪化しやすかったのではないかと考えられる．反対に，単語の削除操作の Perplexity が横ばいになったのは，そのトピックに関連のない単語が無理に追加されることはなく，ランダムに選択されたトピック内の上位に挙げられた単語が消えるだけであるためであると考えられる．下位に挙げられている比較的関連のある単語が上位に繰り上がることによって， Perplexity に影響を大きく与えることなく操作を加えることができた．

また，今回の実験では， Perplexity と反映率を評価指標として用いて比較実験を行った．今回の実験では， HL-NTM 以外のトピックモデルと Perplexity を比較することによってモデルの予測精度が比較的悪化していないことを示し，反映率によって Human-in-the-Loop がうまく機能していることを示した．しかし，これらの評価指標のみでは，実際に人間にとって好ましいスコアを得ることができたと断定することは難しい．なぜならば，どちらも人間にとって好ましい結果を出力できているかを直接，定量的に評価するための指標ではないからである．例えば，選出した単語がそのトピックに占める割合を考慮して学習を行うことでより好ましい結果が得られるような HL-NTM を構築することができるのではないかと考えられる．したがって，人間が求める結果に適応できたということを正確に評価するために，より適したモデルの構築や評価指標を模索する必要がある．

6 まとめと今後の展望

本研究は， NTM と Human-in-the-Loop を組み合わせた Human-in-the-Loop ニューラルトピックモデルを構築し，既存のトピックモデルと比較実験を行い， HL-NTM の有用性が確認できることを目指した．実験では，評価指標として Perplexity と反映率を用いて，比較評価実験を行った．結果として， LDA よりも優れた Perplexity であり， NTM と比較しても Perplexity を大きく悪化させず， Human-in-the-Loop として単語の追加操作と単語の削除操作をランダムで行う仕組みを NTM に加えることが可能であることを示した．

今後の HL-NTM で必要となる操作としては，語順の入れ替え，トピックの結合，トピックの分割，トピックの作成，文書の削除，文書の追加のような操作である．ユーザの操作選択肢を増やし，より人間の求めるトピックが抽出されるようにするには，これらの追加実装と実験が必要不可欠である．また，これらの操作を増やすにあたって，操作の組み合わせや操作の順番によってスコアが変化する可能性も新たに考えられる．

本研究ではランダムなユーザを想定した実験を行った．これに関して，本研究に実装可能である既存手法のシミュレーショ

ン実験について述べる．既存の HL-TM の手法には，Human-in-the-Loop を組み込む構造として，ランダムな操作を行うユーザだけではなく，人間にとって好ましいと考えられる操作を加えるように傾向を与えた「グッドユーザ」を定義してシミュレーション実験を行っているものが存在する [11]．よって，実際に人間による Human-in-the-Loop の操作を加えた実験を行う前実験として，このようなシミュレーション実験を行った後に，人間によるフィードバックを交えた実験を行うことがより好ましいと考えられる．

また，人間による Human-in-the-Loop の操作を加えた実験を行う場合についての考えを述べる．今回の実験では，ランダムに操作を行うユーザを想定した実験を行ったが，今後は，HL-NTM の実用性を視野に入れると，自然言語処理系の専門家を対象のユーザとして HL-NTM に操作をした場合との比較実験も行う必要がある．人間が操作するにあたって，ユーザフレンドリーな UI を設計することが，ヒューマンエラーによる誤操作を削減して，より正確なフィードバックを得ることができる可能性が考えられる．そのため，[13] で紹介されている UI ように，モデルがどのような出力して学習しているのかが理解しやすく，そのプロセスを追跡・変更できるような柔軟な機能と UI が求められる．

このように，シミュレーション実験と実際に人間による Human-in-the-Loop の操作を加えた実験のスコアを比較して実験を行うことによって，性能を評価する 1 つの指標になると考えられる．

謝　　辞

本研究の一部は JSPS 科研費 (JP22H00508，JP23H03405)，JST CREST(JPMJCR22M2) の支援を受けたものである．ここに謝意を示す．

文　　献

[1] D. Magatti, M. Faini, and F. Stella. A software system for topic extraction and document classification. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Vol. 1, pp. 283–286, Los Alamitos, CA, USA, sep 2009. IEEE Computer Society.

[2] M. Wang and P. Mengoni. How pandemic spread in news: Text analysis using topic model. In *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pp. 764–770, Los Alamitos, CA, USA, dec 2020. IEEE Computer Society.

[3] T. Chen, J. Liu, B. Cao, Z. Peng, Y. Wen, and R. Li. Web service recommendation based on word embedding and topic model. In *2018 IEEE Intl Conf on Parallel amp; Distributed Processing with Applications, Ubiquitous Computing amp; Communications, Big Data amp; Cloud Computing, Social Computing amp; Networking, Sustainable Computing amp; Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pp. 903–910, Los Alamitos, CA, USA, dec 2018. IEEE Computer Society.

[4] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, Vol. 41, No. 6, pp. 391–407, 1990.

[5] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57, 1999.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, Vol. 3, No. Jan, pp. 993–1022, 2003.

[7] Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering discrete latent topics with neural variational inference. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 of *Proceedings of Machine Learning Research*, pp. 2410–2419. PMLR, 06–11 Aug 2017.

[8] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models, 2017.

[9] Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, and Shuming Shi. Topic-aware neural keyphrase generation for social media language, 2019.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[11] Varun Kumar, Alison Smith-Renner, Leah Findlater, Kevin Seppi, and Jordan Boyd-Graber. Why didn't you listen to me? comparing user control of human-in-the-loop topic models, 2019.

[12] Khan Muhammad Haseeb Ur Rehman and Kei Wakabayashi. Keyphrase-based refinement functions for efficient improvement on document-topic association in human-in-the-loop topic models. *Journal of Information Processing*, Vol. 31, pp. 353–364, 2023.

[13] Zheng Fang, Lama Alqazlan, Du Liu, Yulan He, and Rob Procter. A user-centered, interactive, human-in-the-loop topic modelling system. *arXiv preprint arXiv:2304.01774*, 2023.

[14] John Lafferty and David Blei. Correlated topic models. *Advances in neural information processing systems*, Vol. 18, , 2005.

[15] Thomas Griffiths, Mark Steyvers, David Blei, and Joshua Tenenbaum. Integrating topics and syntax. *Advances in neural information processing systems*, Vol. 17, , 2004.