DEIM Forum 2022 H21-2

データのスパース性を考慮した企業推薦手法の提案

福知 侑也† 馬 強†

† 京都大学大学院情報学研究科 〒 606–8501 京都府京都市左京区吉田本町 36-1
E-mail: †fukuchi@db.soc.i.kyoto-u.ac.jp, ††qiang@i.kyoto-u.ac.jp

あらまし 近年転職活動をはじめ，キャリア形成や人事採用を支援するシステムやサービスが開発され，特にオンライン採用プラットフォームが活用されている．オンライン採用プラットフォームは多くのユーザが利用しているが，膨大な情報の中から自分にあった転職先企業を探す必要があり，ユーザの負担が大きい．そのため検索や推薦など情報システムを利用した意思決定支援システムの利用が有効とされている．企業・職業推薦の既存手法の多くは大量のユーザ情報やキャリアパスの情報の使用を前提としており，データスパース性が顕著である企業推薦では改善の余地がまだ大きい．そこで本研究ではデータスパース性にアプローチした企業推薦システムを提案する．企業推薦におけるデータスパース性を解消するための企業表現の学習手法，系列推薦モデルとデータの増殖手法を提案し，その有効性を検証した．

キーワード 情報推薦， グラフ， 職業， シーケンシャル推薦， 機械学習

1 はじめに

近年従業員が短期間で転職する傾向が見られ，米国労働統計局 [1] によるとアメリカ人の在職年数の中央値は 2012 年時点で 4.6 年であったが，2020 年時点では 4.1 年となり，減少傾向にある．また日本は他国と比べ勤続期間が長い傾向にあるが，終身雇用制度の衰退などの影響で，人材の流動性が高まっている．「人生 100 年時代」という言葉が注目を浴びており，終身雇用制度も希薄になっている今，多くの人にとって自律的なキャリア形成が重要である．

そのためキャリア形成や人事採用を支援するシステムやサービスが活用されており，特にオンライン採用プラットフォームが果たす役割はますます大きくなっている．有名なオンライン採用プラットフォームである Linkedin [2] では約 8 億人の登録者，約 2000 万件の求人情報の掲載がされている [3]．オンライン採用プラットフォームは多くのユーザが利用しており，転職先企業発見を目的として，膨大な情報の中から自分にあった企業を探す必要があり，ユーザの負担が大きい．検索や推薦など情報システムを利用した意思決定支援システムがますます重要となってくる．

職業推薦 (企業推薦) の既存手法として，ユーザのプロフィール情報または履歴書の情報からいくつかの特徴量を抽出し，それらの特徴量を用いて機械学習モデルを学習し，転職先の企業の推薦を行う手法が多く提案されている [4] [5] [6]．また，Meng ら [7] はユーザごとの過去の就業企業の系列，各企業の役職の系列とユーザの属性情報を用いて階層的に学習する系列推薦モデル HCPNN を提案している．HCPNN はアテンション機構を用いることで推薦における重要な要因を分析するとともに，階層的な系列の学習により系列を考慮しない手法に比べ高い性能を示している．しかしながら既存手法は大量のユーザ情報とキャリアパスを利用しているため，データスパース性が顕著である企業推薦では改善の余地がまだ大きい．

企業推薦におけるデータスパース性は以下の 3 つがあると考える．

- ユーザ情報のスパース性: 近年プライバシ問題により年齢や性別といったユーザの属性情報の収集，利用が困難になっている．また企業としてもユーザの個人情報の利用は物議を醸している．このような背景から個人情報の利用は制限され，推薦モデルの構築に利用可能なキャリアパスの数も制限される．
- 企業情報のスパース性: いわゆる大企業は従業員数が多いことからさまざまなユーザの在職データがある一方で．ベンチャー企業，中小企業はユーザの在職データが少ない．また，企業によって公開される情報のばらつきも大きく，データが不十分である中小企業や新規企業がまだ多い．
- キャリアパスのスパース性: クリック履歴や商品購入履歴など系列推薦で用いられるユーザの行動履歴のデータの性質と比較して，キャリアパスは系列長が短いという課題がある．系列推薦において系列長が短いという問題は，推薦性能低下につながるということが知られている. [8]

そこで，本研究ではこれらのデータスパース性にアプローチした企業推薦システムを提案する．提案する企業推薦システムは企業表現の学習手法，系列推薦モデル，データの増殖手法からなる．

- 企業表現の学習手法: 企業情報・ユーザ情報のスパース性に対処する，企業表現の学習手法を提案する．外部データから収集した企業の属性情報とユーザ・企業グラフのエンベッディング手法を用いて，企業のベクトル表現を学習する．
- 系列推薦モデル: 企業情報のスパース性に対処する系列推薦モデルを提案する．入力であるキャリアパスから業種情報のみを抽出した系列を新たに作成し学習する機構を持つことでデータの不均衡性の解消を試みる．
- データの増殖手法: キャリアパスのスパース性に対処する，データの増殖手法を提案する．キャリアパスデータは系列

の数が少なく，系列の長さが短いことが課題であるため，長さと数の観点からデータを増殖する手法を提案し，各手法の比較分析を行う．

本論文の構成は以下の通りである．第 2 節では関連研究について記述する．第 3 節は基本事項とし，問題定義や使用するデータセットについて記述する．第 4 節では提案手法の詳細について記述する．第 5 節では実験結果と考察を記述する．最後に第 6 節でまとめる．

2 関 連 研 究

2.1 キャリアパスの系列性を考慮しない推薦手法

Snorre ら [6] は GBDT(Gradient-Boosted Decision Trees) モデルを用いて，ユーザ情報を特徴量としてネクスト職業の推薦を行っている．この研究では，2012 年から 2016 年のデンマーク人の背景情報を含む採用市場データを用いている．特徴量として用いるユーザ情報には過去 5 つの企業や勤続年数などの 100 以上のユーザ情報が含まれる．推薦モデルには XGBoost [9] を利用し，ネクスト職業推薦で MRR@10 と Recall@10 の二つの評価指標において，ベースラインの精度を上回ったことを報告している．

2.2 キャリアパスの系列性を考慮した推薦手法

Meng ら [7] は転職先企業の推薦とその企業での在籍期間を予測するため，Hierarchical Career-Path-Aware Neural Network(HCPNN) 法を提案している．HCPNN の特徴はユーザごとに過去の就業企業の系列と各企業の役職の系列を階層的に学習していることである．モデルへの入力としてユーザごとの企業の系列，役職の系列，ユーザの属性情報を用いている．系列学習モデルは LSTM とアテンション機構からなる．処理の順序としては初めに役職のベクトル系列を LSTM に入力し，隠れ層の出力を得る．その出力を企業ごとに結合したベクトルと企業のベクトル系列を結合し，二段目の LSTM モデルに入力し，隠れ層の出力を得る．最後にその出力とユーザの属性情報ベクトルを結合しアテンション機構により次の企業とその在職期間の予測を行う．アテンション機構を用いた系列学習により既存手法に比べ高い性能を示している．しかしながら HCPNN を含む既存研究は大量のユーザ情報を利用しているため，データスパース性が顕著である企業推薦では改善の余地がまだ大きいと考える．

3 基 本 事 項

本節では，提案手法や実験の導入として，問題定義やデータセットに関する基本事項を述べる．はじめに本研究で対象とする問題の定義について記述する．次に本研究で使用するデータセットについて記述する．最後に実サービスを用いたデータセットの補強について述べる．

3.1 問 題 設 定

企業推薦システムの問題設定について述べる．$\boldsymbol{U}$ をユーザ集合，$\boldsymbol{C}$ を企業集合とする．それぞれのユーザ $u \in \boldsymbol{U}$ は一つのキャリアパス $S(u) = \{c_1, c_2, \cdots, c_n\}$ を持つ．ここで $n = |S(u)|$ であり，$c \in \boldsymbol{C}$ は企業を表す．各企業は説明文，業種，人数規模の情報を持つ．また，ユーザ u のキャリアパスは各企業の就業開始時期の昇順で並ぶ．系列推薦ではキャリアパス $S(u)$ を入力に，次の時間ステップにおける全企業に対する就業確率を出力し，上位 k 件の企業を推薦企業として提示する．

$$P\left(c_{|S(u)|+1} = c_i \mid S(u)\right) (c_i \in \boldsymbol{C}) \tag{1}$$

表 1　データセットに含まれる各カラムの値の例

ユーザ ID	企業名	始業開始時期
u1	c1	2010-1-1
u1	c2	2011-1-1
u1	c3	2012-1-1
u2	c2	2010-10-10
u2	c4	2015-1-1
u3	c2	2013-4-5

3.2 データセット

本研究では世界最大規模のビジネス SNS である Linkedin [2] からユーザのプロフィールを収集した kaggle 上の公開データセット [10] を使用する．このデータセットは 2018 年 1 月 1 日時点で収集されたデータである．オーストラリアが国籍であるユーザのこれまでの就業企業や出身大学とその始業開始時期等が含まれる．本研究ではデータセット中のユーザ ID，企業名，始業開始時期の 3 つのカラムを使用した．データの形式の例を表 3. 2 に示す．ユーザ ID は Linkedin におけるユーザ識別子である．このデータセットは計 39,535 行からなり，総企業数は 13,755，総ユーザ数は 6,853 であった．

3.3 データセットの補強

データセットは企業の属性情報を含まないため，本節では企業データベースである Crunchbase [11] を用いてデータセット中の各企業の属性情報を収集した．Crunchbase 上での’Crunchbase’の企業ページを図 1 に示す．本研究で扱うデータセット中の 13755 件の企業のうち，13657 件の情報を Crunchbase から収集した．図 1 にもみられるように，収集したデータには業界タグ，地域，企業の設立日，企業の説明文，従業員数が含まれ，本研究では業界タグ，企業の説明文，従業員数を用いた．業界タグは文献 [12] のように Crunchbase 上で定義されており，743 の業界タグと業界タグをグループ化した 47 の業界グループがある．また，企業は複数の業界タグを持つ．企業の説明はその企業のサービスや業態を英語による自然言語で記述している．従業員数は「1-10 人」,「11-50 人」,「51-100 人」,「501-1000 人」,「1001-5000 人」,「5001-10000 人」,「10001 人以上」の区分のいずれかが設定されている．

図 1　Crunchbase 上での企業ページの例

4　企業推薦手法

4.1　提案手法の概要

本研究ではデータのスパース性を解決する企業推薦手法を提案する．以下の三つのデータスパース性に着目する．

- ユーザ情報のスパース性: 近年ユーザの企業へのデータ提供の不安や企業の個人情報の利用といったプライバシ問題により年齢や性別といったユーザの属性情報の収集，利用が困難になっている．このような背景から個人情報の利用は制限され，学習に利用可能なキャリアパスの数も制限されてしまう．
- 企業のスパース性: いわゆる大企業は従業員数が多いことからさまざまなユーザの在職データがある一方で．ベンチャー企業，中小企業はユーザの在職データが少ない．学習データに偏りがあることで系列推薦モデルは出現頻度の高い企業を強く学習してしまう．
- キャリアパスのスパース性: クリック履歴や商品購入履歴など系列推薦で用いられるユーザの行動履歴のデータの性質と比較して，キャリアパスは系列長が短いという課題がある．系列推薦において系列長が短いという問題は，推薦性能低下につながるということが知られている．[8]

提案手法は三つの手法から構成される．提案手法の全体像を図 2 に示す．

- 企業の表現学習: 外部データから収集した企業の属性情報と企業-ユーザグラフを用いて，企業のベクトル表現を学習する手法を提案する．企業とユーザの関係を学習し企業のベクトル表現を得ることで企業情報のスパース性，ユーザ情報のスパース性に対処する．
- 系列推薦モデル: 企業情報のスパース性に対処する系列推薦モデルを提案する．入力であるキャリアパスから業種情報のみを抽出した系列を新たに作成し学習する機構を持つことでデータが不均衡性の解消を試みる．また，時系列性を持つデータの学習に有効な LSTM モデルにデュアルアテンション機構を導入して系列推薦を行う．
- データ増殖: キャリアパスのスパース性に対処する，データの増殖手法を提案する．キャリアパスデータは系列の数が少なく，系列の長さが短いことが課題であるため，長さと数の観点からデータを増殖する手法を提案する．具体的には，系列数を増やす操作として Mask 処理，Crop 処理，Substitute 処理，Reorder 処理を，系列長を増やす操作として Insert 処理，在職期間を考慮した Insert 処理を定義し，それらを用いてより多様な長い系列を生成する．

4.2　企業表現の学習手法

本節では企業のベクトル表現を学習する手法について記述する．企業-ユーザグラフのエンベッディング手法により企業とユーザの関係，企業間の関係を学習し企業のベクトル表現を得ることで企業情報のスパース性，ユーザ情報のスパース性に対処する．企業-ユーザの二部グラフを用いる目的は以下である．

- ユーザと企業間の関係を企業のベクトル表現に反映する．
- 企業のスパース性の問題に対処するため 3.3 節で示したように，データセットの拡張を行った．しかし各企業の属性情報のみでは企業間の関係を考慮したエンベッディングが難しいため，グラフエンベッディング手法により企業間の関係を考慮し情報を補完する．
- 企業の属性情報の一部には欠損が見られるため，情報の欠損を企業間関係の学習により補完する．

はじめに企業の属性情報を用いたエンベッディング手法について述べ，次に二部グラフの学習によるエンベッディング手法について述べる．

4.2.1　企業の属性情報を用いたエンベッディング

企業の属性情報として Crunchbase から取得した各企業の説明文，業界情報と従業員数を用いる．英語表現の企業の説明文を Sentence-BERT [13] を用いて 1024 次元のベクトル表現を得る．ここで得る企業 c の文章ベクトルを $\boldsymbol{T}(\vec{\boldsymbol{c}})$ とする．業界情報に関しては，Crunchbase が定義する業界グループ情報を用いて，multihot-encoding により 47 次元のベクトルに変換した．ここで得る企業 c の業界ベクトルを $\boldsymbol{I}(\vec{\boldsymbol{c}})$ とする．従業員数は 3.3 節で示したように 7 つの区分で設定されているため onehot-encoding により 7 次元のベクトルに変換した．このベクトルを $\boldsymbol{E}(\vec{\boldsymbol{c}})$ とする．企業ベクトルは上記のベクトルを結合した

$$\vec{\boldsymbol{v}_c} = \{\boldsymbol{I}(\vec{\boldsymbol{c}}), \boldsymbol{T}(\vec{\boldsymbol{c}}), \boldsymbol{E}(\vec{\boldsymbol{c}})\} \tag{2}$$

とする．

4.2.2　企業-ユーザグラフのエンベッディング

企業ノードとユーザノードの 2 種類のノードを持つ 2 部グラフ $\mathcal{G} = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{E})$ を作成する．ここで $\boldsymbol{U}$ はユーザノードの特徴ベクトルの集合，$\boldsymbol{V}$ は企業ノードの特徴ベクトルの集合，$\boldsymbol{E}$ はユーザノードと企業ノード間のエッジの集合である．ユーザベクトルについては 0 ベクトルで初期化を行なった．ユーザノードと企業ノードの間にエッジが存在することはそのユーザが過去にその企業に所属していたことを表す．

作成した 2 部グラフ $\mathcal{G}$ を用いてリンク予測タスクを解くこと

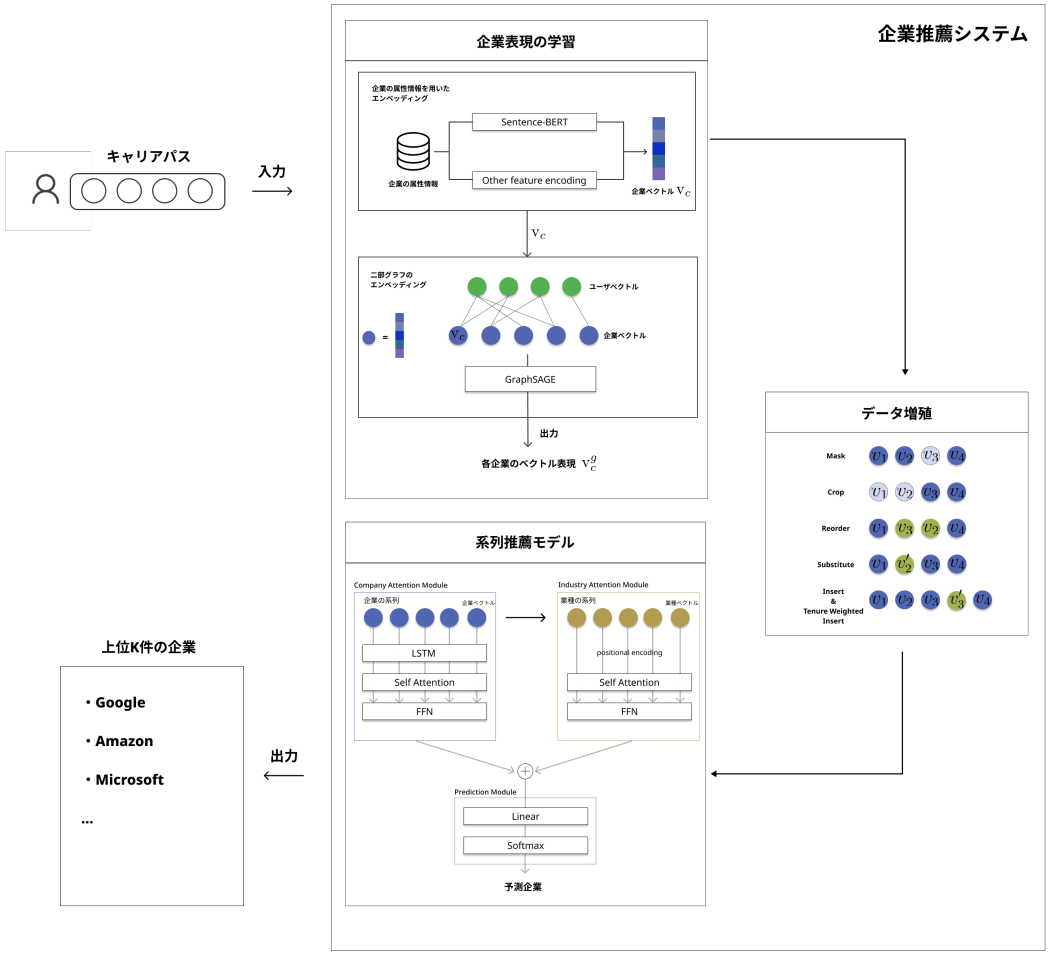

図 2　提案手法の概要

で各ノードのベクトル表現を得る．リンク予測とはグラフ内の 2 つのノード間のエッジの存在を予測する二値分類問題である．この問題を解くために GraphSAGE [14] を用いた．学習データの作成に関しては，ユーザノードと企業ノード間に存在するエッジをランダムに l 個選択し，存在しないエッジをランダムに同数の l 個選択し学習データを作成した．企業 c の学習後の企業ベクトルを $\vec{v_c^g}$ とする．

4.3　系列推薦モデル

企業の出現頻度分布はロングテールな分布になっている．いわゆる大企業は従業員数が多いことから多くのユーザの在職データがある一方で，ベンチャー企業，中小企業はユーザの在職データが少ない．学習データに偏りがあることで系列推薦モデルは出現頻度の高い企業を強く学習してしまう．そこでキャリアパスから業種情報のみを抽出した系列を作成し学習する機構を持つ Dual Attention Network を提案する．Dual Attention Network の概略図を図 3 に示す．提案モデルは Company Attention Module と Industry Attention Module と Prediction Module からなる．

4.3.1　Company Attention Module

このモジュールでは企業のベクトルの系列 $S(u) = \{\vec{v_1^g}, \vec{v_2^g}, \cdots, \vec{v_n^g}\}$ を LSTM の入力とし出力 $\boldsymbol{h} = \{h_1, h_2, \cdots, h_n\}$ を得る．LSTM の更新式は以下である．h_t は t 番目のアイテムの隠れ層の状態であり，c_t は t 番目のセル状態である

$$h_t = \mathrm{LSTM}(\vec{\boldsymbol{v_t^g}}, c_{t-1}, h_{t-1}) \tag{3}$$

その後出力 $\boldsymbol{h}$ に対してセルフアテンション機構を用いてこのモジュールの最終的な出力を得る．セルフアテンション層の出力 $\vec{\boldsymbol{a^c}}$ を得る式を式 4 に示す．

$$\vec{\boldsymbol{a^c}} = \mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

ここで $Q = K = V = \boldsymbol{h}$ である．また，d_k はスケール因子であり，ソフトマックス関数による勾配消失を軽減するパラメータである．

4.3.2　Industry Attention Module

このモジュールでは企業の系列から業界の情報を新たな系列として抽出し，セルフアテンション機構を用いて出力としている．企業の系列がスパースであることから，業界情報を切り分けることでスパース性の解消を試みる．このモジュールの入力である業界の系列を $S^I(u) = \{\boldsymbol{I}(\vec{\boldsymbol{c_1}}), \boldsymbol{I}(\vec{\boldsymbol{c_2}}), \cdots \boldsymbol{I}(\vec{\boldsymbol{c_n}})\}$ とし，Attention 層の出力 $\vec{\boldsymbol{a^I}}$ を得る式を以下に示す．

$$\vec{\boldsymbol{a^I}} = \mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

ここで $Q = K = V = S^I(u)$ である．

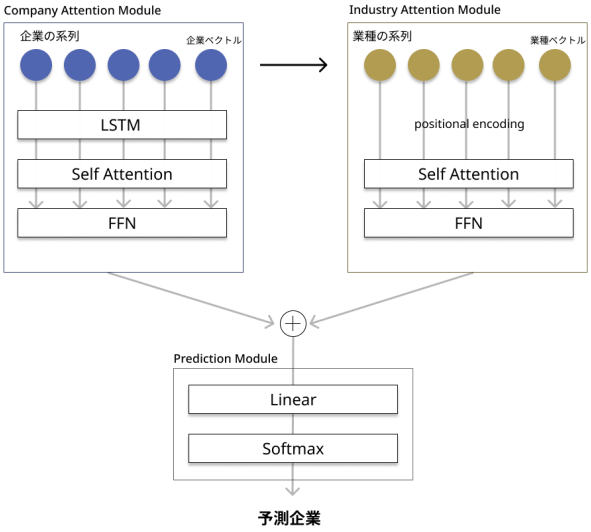

図 3　Dual Attention Network

4.3.3　Prediction Module

この層では Company Attention Module と Industry Attention Module の出力である $\vec{\boldsymbol{a}^c}, \vec{\boldsymbol{a}^I}$ を結合し，全結合層と Softmax 関数を経て多クラス分類を行う．$\vec{\boldsymbol{a}^{concat}} = \{\vec{\boldsymbol{a}^c}, \vec{\boldsymbol{a}^i}\}$ とし，出力を得る式を式 6 に示す．

$$\mathrm{P}(c_{n+1}|S(u)) = \mathrm{Softmax}(\mathrm{Linear}(\vec{\boldsymbol{a}}^{concat})) \tag{6}$$

損失関数にはクロスエントロピー関数を用いた．

4.4　データの増殖手法

キャリアパスデータは数が少なく，長さが短いことが課題である．そこで本研究では系列の数と長さの観点からデータを増殖する手法を提案する．系列数を増やす増殖手法として Crop, Mask, Reorder, Substitute [15] の 4 つのデータ増殖手法を用いる．系列長を増す増殖手法として Insert，TenureWeightedInsert の 2 つのデータ増殖手法を提案する．

4.4.1　系列数の増殖手法

以下では増殖する対象となる系列を $s_u = \{v_1, v_{i+1}, \cdots, v_n\}$ とし系列数を増やす増殖手法である Crop, Mask, Reorder, Substitute の 4 つのデータ増殖手法を説明する．

- Crop: ランダムに系列の一部をサブ系列として抽出する．Crop 処理後の系列 s_u^M を以下に示す．

$$s_u^M = \{v_i, v_{i+1}, \cdots, v_{i+c-1}\} \tag{7}$$

c は Crop 後の系列の長さである．サブ系列の長さを決定するパラメータ $\eta(0 < \eta < 1)$ を用いて，$c = \eta n$ とする．i はクロップするサブ系列の開始インデックスであり，$0 < i < n - c$ である．

- Mask: ランダムに系列の要素の中から l 個の要素を選択しマスク処理をする．Mask 処理後の系列 s_u^M を以下に示す．

$$s_u^M = \{v'_1, v'_2, \cdots, v'_n\} \tag{8}$$

系列の何割の要素をマスクするかを決定するパラメータ $\gamma(0 < \gamma < 1)$ を用いて $l = \gamma n$ とする．v'_i はマスク後の要素であり，選択されなかったものについては $v'_i = v_i$ である．

- Reorder: 系列から連続するサブ系列を選択しシャッフルする．Reorder 処理後の系列 s_u^R を以下に示す．

$$s_u^R = \{v_1, v_2, \cdots, v'_i, \cdots, v'_{i+r-1}, \cdots, v_n\} \tag{9}$$

$\{v'_i, \cdots, v'_{i+r-1}\}$ はシャッフル後のサブ系列である．対象となるサブ系列の長さを決定するパラメータ $\omega(0 < \omega < 1)$ を用いて，サブ系列の長さ $l = \omega n$ とする．

- Substitute: 企業間類似度を用いてランダムに系列の要素の一部をその要素と類似する要素に置換する．具体的には，系列 s_u からランダムに k 個のインデックスの集合 $\{idx_1, idx_2, \cdots, idx_k\}$ を選択し，類似する要素に置き換える．Substitute 処理後の系列 s_u^S を以下に示す．

$$s_u^S = \{v_1, v_2, \cdots, v'_{idx_1}, \cdots, v'_{idx_k}, \cdots, v_n\} \tag{10}$$

ここで v'_{idx} は v_{idx} と最も類似度が高い要素である．系列の要素の何割を置き換えるかを決定するパラメータ $\mu(0 < \mu < 1)$ を用いて，$k = \mu n$ とする．

Crop, Mask, Reorder はランダムな増殖手法であり，対象となる要素の性質を考慮しない．一方 Substitute は対象となる要素と別の要素との類似度を用いた増殖手法である．要素の類似度としてはコサイン類似度を用いた．n 次元の要素 $\boldsymbol{x}$ と $\boldsymbol{y}$ のコサイン類似度は以下で計算する．

$$\cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

4.4.2　系列長の増殖手法

系列の要素の特徴を考慮しながら系列長を増し，パディング処理の役割も担う手法を提案する．系列長を増す増殖手法として Insert, TenureWeightedInsert の 2 つの手法を提案する．以下では増殖する対象となる系列を $s_u = \{v_1, v_{i+1}, \cdots, v_n\}$ とし各データ増殖手法を説明する．

- Insert: 企業間類似度を用いて系列に要素を挿入する．具体的には，系列 s_u からランダムに k 個のインデックスの集合 $\{idx_1, idx_2, \cdots, idx_k\}$ を選択し，選択したインデックスに要素を挿入する．Insert 処理後の系列 s_u^I を以下に示す．

$$s_u^I = \{v_1, v_2, \cdots, v_{idx_1}, v'_{idx_1}, \cdots, v_{idx_k}, v'_{idx_k}, \cdots, v_n\} \tag{11}$$

ここで v'_{idx} は v_{idx} と最も類似度が高い要素である．挿入後の系列長は $n + k$ である．系列の要素の何割を挿入するかを決定するパラメータ $\zeta(0 < \zeta < 1)$ を用いて，$k = \zeta n$ とする．

- TenureWeightedInsert: ユーザのある企業への在職期間はユーザの嗜好を表すのに重要であると考える．在職期間が長い企業がユーザにとって重要であるという仮説を立て，在職期間で重み付けをして系列に要素を挿入する手法を提案する．具体的には，系列 s_u に対応する在職期間の系列を $T_u = \{t_1, t_2, \cdots, t_n\}$ とし i 番目のインデックスに $t_i/\Sigma t_k$ の確率で要素を挿入する．挿入処理後の系列 s_u^T を以下に示す．

$$s_u^T = \{v_1, v_2, \cdots, v_{idx_1}, v'_{idx_1}, \cdots, v_{idx_k}, v'_{idx_k}, \cdots, v_n\} \tag{12}$$

ここで v'_{idx} は v_{idx} と最も類似度が高い要素である．

一方 Insert，TenureWeightedInsert は対象となる要素と別の要素との類似度を用いた増殖手法である．要素の類似度としてはコサイン類似度を用いた．

5 評価実験

本研究では企業推薦のためのスパース性解消手法として企業表現の学習手法，系列推薦モデル，データ増殖手法を提案した．本節ではそれぞれの手法の有効性の検証を行う．

5.1 評価指標

評価指標には Accuracy@K(Acc@K) と Mean Reciprocal Rank(MRR) の二つを用いた．Acc@K はユーザに K 個のアイテムを推薦したとき，正解のアイテムがその K 個に含まれていれば正解とするときの正解率である．以下のように計算する．

$$\text{Acc@ } K = \frac{1}{N}\sum_{l=1}^{N} I(r(l) \leqq K)$$

ここで $r(l)$ は l 番目のユーザに推薦した企業の順位であり，N は予測対象の系列の数，$I(\cdot)$ は条件式が正であれば 1 を，負であれば 0 を返す指示関数である．本実験では $K = 1, 15, 30$ を K の値として用いた．

MRR は最初に現れた適合アイテムの順位の逆数の平均をとったものである．以下のように計算する．

$$\text{MRR } = \frac{1}{N}\sum_{l=1}^{N} \frac{1}{r(l)}$$

ここで $r(l)$ は l 番目のユーザに推薦した企業の順位であり，N は予測対象の系列の数である．Acc@K と MRR はどちらも値が大きいほど，良い推薦システムであることを示す．

5.2 企業表現の学習手法の検証

二部グラフ (BG) によるエンベッディングの有効性の検証をするために，二部グラフでのエンベッディングの有無による性能変化を検証した．二部グラフを用いない場合，各企業のベクトルの表現方法は二部グラフ入力前の企業ベクトル $\vec{v_c}$ とした．結果を表 2 に示す．表より，4 つの評価指標で二部グラフを用いた場合の推薦性能が上回っており，本研究における問題設定では二部グラフによるエンベッディングが有効であることがわかる．二部グラフによってユーザと企業，企業間の関係を学習し，また企業の属性情報の欠損を補完することでより良い企業表現を学習したと考える．

表 2　二部グラフの有効性の検証

	Acc@1	Acc@15	Acc@30	MRR
Our Model w/o BG	0.0104	0.0847	0.1248	0.0327
Our Model	0.0127	0.0927	0.1274	0.0349

5.3 系列推薦モデルの有効性の検証

本節では系列推薦モデルの有効性の検証を行う．はじめに LSTM モデルと提案モデルの性能比較を行う．次に提案モデルにおける Industry Attention Module の有効性を検証する．また，学習データ量による性能変化を検証するため，各訓練データ比率による評価指標への影響を検証する．最後にキャリアパスの系列長による性質を調べるため系列長ごとの性能比較を行う．

5.3.1 Dual Attention Network と LSTM の性能比較

LSTM モデルと提案モデルの性能比較を行う．結果を表 4 に示す．表 4 より 4 つの評価指標において提案モデルは LSTM を用いた場合に比べて性能が良いことがわかった．

表 3　Dual Attention Network と LSTM の性能比較

	Acc@1	Acc@15	Acc@30	MRR
LSTM	0.0097	0.0812	0.1161	0.0307
Dual Attention Network	0.0127	0.0927	0.1225	0.0350

5.3.2 Industry Attention Module の有効性の検証

Dual Attention Network の構成要素の一つである業界情報を切り分けて学習する Industry Attention Module の有効性を検証した．比較対象として我々のモデルから Industry Attention Module を取り除いたものと比較した．結果を表 3 に示す．表より，Industry Attention Module を用いた結果が上回っていることがわかる．

データセット中の企業の出現回数による不均衡性によって，系列推薦モデルは出現頻度の高い企業を強く学習してしまう．Industry Attention Module を用いることでこの問題に対処していると考えられる．

5.4 データの増殖手法の有効性の検証

本研究では系列長を増やすことと系列数を増やすことを目的に 6 つのデータ増殖手法を提案した．本節でははじめに 6 つの系列数増殖手法の性能比較を行う．次に系列数増殖手法と系列長増殖手法の組み合わせについて検証する．

5.4.1 各データ増殖手法の性能比較

提案した 6 つの系列数増殖手法の効果を検証する．系列の最大長 $T = 5$ とし，各増殖手法”Crop”, ”Mask”, ”Reorder”, ”Substitute”, ”Insert”, ”TenureWeightedInsert”を用いた結果を表 5 に示す．また，データ増殖後の系列推薦モデルとして Dual Attention Network を用いた．

表 5 よりデータ増殖を用いない場合に比べて，いずれかのデータ増殖を行なった場合の性能が高いことがわかる．Acc @ 1 と MRR の指標においては”Insert”の増殖手法を用いたものが最も性能が良く，Acc @ 15 と Acc@30 の指標において

表 4　Industry Attention Module の有効性の検証

	Acc@1	Acc@15	Acc@30	MRR
Dual Attention Network w/o industry attention module	0.0116	0.0909	0.1107	0.0326
Dual Attention Network	0.0127	0.0927	0.1225	0.0350

表 5　各系列増殖手法の性能比較

	増殖手法	Acc@1	Acc@15	Acc@30	MRR
	増殖なし	0.0127	0.0927	0.1274	0.0349
系列数増殖手法	Mask 手法で増殖	0.0139	0.093	0.131	0.0357
	Crop 手法で増殖	0.0153	0.093	0.1274	0.0369
	Reorder 手法で増殖	0.0170	**0.0965**	**0.1312**	0.0379
	Substitute 手法で増殖	0.0163	0.0951	0.1272	0.0378
系列長増殖手法	Insert 手法で増殖	**0.0179**	0.0949	0.1291	**0.0385**
	TenureWeightedInsert 手法で増殖	0.0165	0.0963	0.1295	0.0376

表 6　系列長ごとの各増殖手法の性能比較

系列長 $l \leqq 2$				
	Acc@1	Acc@15	Acc@30	MRR
増殖なし	0.0142	0.0982	0.1302	0.0385
Mask 手法で増殖	0.0118	0.0959	0.1314	0.0368
Crop 手法で増殖	0.0142	0.097	0.1302	0.0395
Reorder 手法で増殖	0.0166	0.0947	0.1243	0.0391
Substiute 手法で増殖	0.013	0.103	0.1219	0.0398
Insert 手法で増殖	0.0154	0.097	0.1195	0.0398
TenureWeightedInsert 手法で増殖	0.0166	0.097	0.1219	0.0407
系列長 $2 < l \leqq 5$				
	Acc@1	Acc@15	Acc@30	MRR
増殖なし	0.0124	0.0914	0.1267	0.034
Mask 手法で増殖	0.0144	0.0922	0.1309	0.0355
Crop 手法で増殖	0.0156	0.092	0.1267	0.0363
Reorder 手法で増殖	0.0171	0.097	0.1329	0.0377
Substiute 手法で増殖	0.0171	0.0931	0.1285	0.0373
Insert 手法で増殖	0.0186	0.0943	0.1314	0.0382
TenureWeightedInsert 手法で増殖	0.0165	0.0961	0.1314	0.0368

は”Reorder”の増殖手法を用いたものが性能が良くなった．

次に系列長による増殖手法の影響を検証するため系列長 $l \leqq 2$ と $l > 2$ にデータを分けて評価した．結果を表 6 に示す．表 6 より”Mask”や”Crop”手法は系列長が短い場合 ($l \leqq 2$ の場合) には有効ではないことがわかる．理由として”Mask”手法は系列の長さを削減する増殖手法であるため，情報量を減らしてしまうことが考えられる．一方”Insert”や”Tenure Weighted Insert”手法は系列長が短い場合にも有効であることがわかる．これら 2 手法は系列長を増やす手法であり，データ増殖により情報を補完することで精度向上につながっていると考えられる．また，系列長が比較的長い場合 ($l > 2$ の場合) においては全てのデータ増殖手法が有効であることがわかる．多様な系列を作成することで精度向上につながっていると考えられる．

5. 4. 2　データ増殖手法の組み合わせ処理の検討

本研究では系列数を増やす 4 つの増殖手法と系列長を増やす 2 つの増殖手法を紹介した．本節では系列数を増やす手法と系列長を増やす手法の組み合わせ方及び処理の順序について検討し，比較する．以下ではパディング前の系列長を L' とし，入力の固定長 T とする．組み合わせの戦略には以下の二つの方法が考えられる．

- One Step 戦略: 入力系列が $L' < T$ の場合に Insert 処理によって系列長を T に揃える．入力系列が $L' = T$ の場合，系列長を変化させない系列数増殖処理を適用し，系列数を増殖する．この手法の利点はモデルの入力の系列長を固定した場合にパディング処理が不要であることである．
- Two Step 戦略: 入力系列に各系列数増殖手法を適用した後，パディング処理を行う．パディング処理としてゼロパディング処理と提案手法である系列長増殖手法によるパディングが考えられる．この手法の利点はモデルの入力の系列長に応じて柔軟にパディング処理によって系列長を揃えることが可能であることである．

系列推薦モデルの入力の固定長 $T = 5$ とし，以下の手法を比較する．

（1） One Step 戦略: 入力系列が $L' < T$ の場合に Insert 手法によって系列長を T に揃える．入力系列が $L' = T$ の場合，Reorder 手法または Substitute 手法を適用し，系列数を増殖する．

（2） Two Step 戦略 (ゼロパディング): Reorder 手法を適用し，系列数を増殖した後，ゼロパディングを行い入力とする．

（3） Two Step 戦略 (Insert 手法によるパディング): Reorder 手法を適用し，系列数を増殖した後，Insert 手法によるパディングを行い入力とする．

表 7 はこれらの手法を比較した結果である．One Step 戦略では $L' <= 4$ で Insert 手法を適用し，$L' = 5$ の場合に系列数増殖手法を適用しないものが最も性能がよいことがわかった．対して，$L' <= 4$ で Insert 手法を適用し $L' = 5$ で Reorder と Substitute 手法を適用したものが最も性能が悪くなった．小節 5. 4. 1 で示したように，Insert 手法は有効であるが，系列長 $L' = 5$ の場合に系列数を増殖する処理によって性能が低下していると考えられる．Two Step 戦略では Insert 手法によるパディング処理はゼロパディング処理を用いた場合より性能が低下した．系列数増殖後に Insert 手法を用いてパディング処理を行うことで，元の系列と大きく異なる系列を作成しているためだと考えられる．One Step 戦略で $L' <= 4$ で Insert 手法を適用し，$L' = 5$ の場合に系列数増殖手法を適用しない手法と Two Step 戦略でゼロパディング処理を用いた手法は近い性能となった．よってこれらの戦略の選択はそれぞれの利点を考慮し，アプリケーションによって臨機応変に選択できる．

まとめとして，小節 5. 4. 1 より Insert，Reorder 処理はそれぞれ有効である．しかし Reorder 手法を適用をした系列に Insert 手法によるパディングを行うことで性能は低下した．また，$L' <= 4$ で Insert 手法を適用し，$L' = 5$ の場合に系列増殖手法を適用した場合も性能が低下した．Reorder 処理と Insert 手法の組み合わせは更なる検証が必要である．

表 7　One Step 戦略と Two Step 戦略の比較

手法	Acc@1	Acc@15	Acc@30	MRR
One Step 戦略 ($L' <= 4$ で Insert 手法を適用)	0.0179	0.0949	0.1291	0.0385
One Step 戦略 ($L' <= 4$ で Insert $L' = 5$ で Reorder 手法を適用)	0.0130	0.0871	0.1227	0.0324
One Step 戦略 ($L' <= 4$ で Insert $L' = 5$ で Substitute 手法を適用)	0.0137	0.0842	0.1225	0.0329
One Step 戦略 ($L' <= 4$ で Insert $L' = 5$ で Reorder と Substitute 手法を適用)	0.0118	0.0847	0.1177	0.0322
Two Step 戦略 (ゼロパディング)	0.0170	0.0965	0.1312	0.0379
Two Step 戦略 (Insert 手法によるパディング)	0.0142	0.0821	0.1159	0.0337

6 ま と め

本研究ではデータのスパース性が顕著である企業推薦において，ユーザ情報のスパース性，企業情報のスパース性，キャリアパスのスパース性にアプローチする企業表現の学習手法，系列推薦モデル，データの増殖手法を提案した．

我々は 13,757 件の企業情報を含む新たな公開可能なデータセットを作成し，提案手法の検証を行った．企業の表現学習の評価実験では二部グラフの有無による性能比較を行い推薦精度 (Mean Reciprocal Rank) が 6.5%改善したことを確認し，今回の問題設定で二部グラフを用いた企業表現の学習が有効であることがわかった．系列推薦モデルの評価実験では LSTM との性能比較を行い，推薦精度 (MRR) が 14%改善した．また，Industry Attention Module の有無による性能比較の結果，Industry Attention Module が有効であるということがわかった．データ増殖手法の評価実験では各データの増殖手法の性能比較を行い，系列長増殖手法では Insert 手法を用いたものが最も性能が良く，系列数増殖手法では Reorder 手法を用いたものが性能が良くなった．また，短い系列に対して系列長の増殖手法が有効であることを確認し，長い系列に対して系列数・系列長の増殖手法が共に有効であることを確認した．しかし系列数増殖手法と系列長増殖手法の組み合わせにより性能低下する場合があり，数と長さの増殖手法の組み合わせは更なる検証が必要である．

7 謝　　辞

本研究の一部は科研費（19H04116）による．

文　　献

[1] Us bureau of labor statistics, 2020. https://www.bls.gov/news.release/tenure.t01.html.
[2] Linkedin, 2022. https://www.linkedin.com/.
[3] Linkedin. About us, 2022. https://news.linkedin.com/about-us#.
[4] Y. Zhang, C. Yang, and Z. Niu. A research of job recommendation system based on collaborative filtering. In *2014 Seventh International Symposium on Computational Intelligence and Design*, Vol. 1, pp. 533–538, 2014.
[5] Ioannis Paparrizos, Berkant Cambazoglu, and Aristides Gionis. Machine learned job recommendation. pp. 325–328, 10 2011.
[6] Snorre S. Frid-Nielsen. Find my next job: Labor market recommendations using administrative big data. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, p. 408–412, New York, NY, USA, 2019. Association for Computing Machinery.
[7] Qingxin Meng, Hengshu Zhu, Keli Xiao, Le Zhang, and Hui Xiong. A hierarchical career-path-aware neural network for job mobility prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, p. 14–24, New York, NY, USA, 2019. Association for Computing Machinery.
[8] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S. Yu. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul 2021.
[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, p. 785–794, New York, NY, USA, 2016. Association for Computing Machinery.
[10] Linkedin profiles and jobs data. https://www.kaggle.com/killbot/linkedin-profiles-and-jobs-data, 2022.
[11] Crunchbase, 2022. https://www.crunchbase.com/.
[12] What industries are included in crunchbase?, 2022. https://support.crunchbase.com/hc/en-us/articles/360043146954-What-Industries-are-included-in-Crunchbase-.
[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.
[14] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.
[15] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S. Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation, 2021.