node2vec を用いた障害レポートにおける故障原因推定

勝又 智[1]　小町 守[1]　真鍋 章[2]　大頭 威[2]　嶋﨑 優子[2]

[1] 首都大学東京 [2] 富士電機株式会社

katsumata-satoru@ed.tmu.ac.jp, komachi@tmu.ac.jp,
{manabe-akira, daitou-takeshi, shimazaki-yuuko}@fujielectric.com

1 はじめに

本研究は，障害レポートにおける**故障状況**に対して，この故障状況に関連する**原因**を適切に推薦する，という問題に取り組む[1]．具体的には，表 1 のように，すでに障害レポートから故障状況，故障原因が抜き出されていて，既知の故障状況とその原因の関連性が事前に分かっている設定である．その中で，新しい故障状況が出現した際に，その故障状況に関連する原因を，いくつかの原因の候補の中から推薦する．

近年，単語分散表現が自然言語処理の様々な場面で使用されている．単語分散表現の学習に共起文脈の分布を使用すると，学習データに対する分布類似度を学習することになり，単語の意味に関する類似度を求めることができる．一方で，分布類似度で捉えることのできない単語間のある関連性を獲得したい場合，単語分散表現ではそのような関係性獲得は難しい．関連する用語の獲得に，単語分散表現ではなく，事前に分かっている故障状況と原因の関連性を考慮した分散表現獲得手法を用いた方が推薦精度が高いのではないか，というのが本研究の仮説である．

本研究では，関連性を考慮した分散表現獲得手法として，単語分散表現をグラフに拡張した Network Embedding 手法，特に node2vec [2] を用いた．我々は故障状況と原因の各単語を節点とした故障状況と原因間の 2 部グラフを構築することで，故障状況とその原因が隣接したグラフを構築した．そしてこのグラフの節点の分散表現として Network Embedding を用いて関連性を考慮した．単語分散表現は単語間の類似性は求めることができるが，関連性を獲得できるかは明らかではない．そのため，この研究が取り組む問題に対しては単語分散表現よりも Network Embedding による節点分散表現獲得が有効であると考えられる．

新規の故障状況に対して関連する原因を推薦する実験を行った結果，Skip-gram に対して node2vec による推薦は，precision@1 で 2.50 ポイント，mean average precision で 1.94 ポイント向上していることを確認した．分野によらず，一貫して node2vec の方が mean average precision において精度が高いことを確認した．

表 1: 障害レポートとそこから抜き出した故障状況と故障原因の例

障害レポート	故障状況	故障原因
冷蔵庫にて**冷却不足**．**クーラー氷結**の為，弱冷．	冷却不足	クーラー氷結
温度異常発報．**ドレンパン氷結**，ケース修理．	温度異常	ドレンパン氷結

[1] この故障状況と故障原因は名詞句を想定している．

2 関連研究

2.1 Network Embedding

Network Embedding には単語分散表現，特に Skip-gram [3] をグラフ構造に対して拡張した手法がいくつか提案されている．これらの手法では，グラフの節点が単語分散表現の学習における単語に対応しており，この節点の分散表現を獲得することを目的としている．Skip-gram は文中の各単語に対して文脈語を決め，ある単語に対して，その文脈語の出力確率を最大化するように学習を行う．同様に，Skip-gram の考えを元にした Network Embedding はグラフ中の各節点に対して文脈節点を決め，ある節点に対してその文脈節点の確率を最大化するように学習を行う．文脈節点は，ある節点を始点とした時のグラフ探索した点である．

このように単語分散表現と Network Embedding の大きな違いの 1 つに文脈語の決め方が挙げられる．Network Embedding ではこの文脈節点を獲得する方法がいくつか研究されている．代表的なものとして node2vec が存在する．node2vec はランダムウォークの探索戦略として幅優先と深さ優先の度合いをハイパーパラメータで制御している．図 1 にその例を示す．

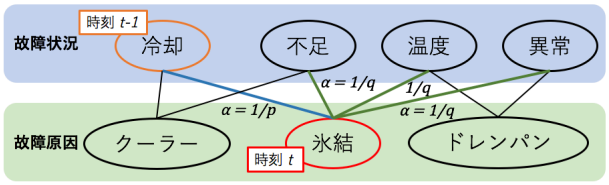

図 1: 表 1 のデータから作成した 2 部グラフ例

ある時刻 t に，節点 v_{t-1}（‘冷却’）から v_t（‘氷結’）へと遷移した状態で，次に探索する節点を考える．次に遷移する節点は 1 つ前の節点からの距離に基づいて，元々の辺の重み[2]にバイアスをかける．探索時のバイアスの値 α は次式のようにして求める．$d(v_{t-1},*)$ は節点 v_{t-1} と，v_t の隣接節点との距離を表す．$*$ は v_t の隣接節点であり，この場合だと ‘冷却’，‘不足’，‘温度’，‘異常’ が該当する．最終的な辺の重みはこの α と元々の辺の重みの積となる．

$$\alpha_{pq}(v_{t-1},*) = \begin{cases} \frac{1}{p} & \text{if } d(v_{t-1},*) = 0 \\ \frac{1}{q} & \text{if } d(v_{t-1},*) = 2 \end{cases}$$

$p, q > 0$ は探索として幅優先なのか深さ優先なのかを制御するハイパーパラメータである．node2vec はこのように文脈節点を近傍から取るのか，深い節点から取るのかをハイパーパラメータで決定している．本研究ではこの node2vec を用いて，分散表現を獲得した．

2.2 自然言語処理への応用事例

この Network Embedding を自然言語処理へ応用した事例がいくつか報告されている．代表的なものとして，コミュニティを考慮した質問応答 [1] への応用が存在する．Zhao ら [4] は，ユーザ，質問，応答の関係性をグラフ構造として考え，ユーザ間のソーシャルな関係性の分散表現を獲得し，質問に対するエキスパートを探す研究を報告している．本研究も故障状況とその原因という関係性を踏まえた分散表現を獲得して故障状況に対応する故障原因の推薦を行うため，Network Embedding を利用している．

3 障害レポートにおける故障状況の原因推定

3.1 問題設定

本研究で使用するデータの形式は，表 1 のように障害レポート，故障状況，故障原因のトリプルである．

[2]本研究では故障状況と原因の共起頻度に基づく重み付きグラフを node2vec の学習に使用している．

本研究の目的は，新規の故障状況が出現した際にその原因を推薦することである．

上記の目的に基づいて，本研究では使用するデータを学習用データと評価用データに分割し，評価用データの故障状況に対して正しい故障原因を推薦できるかを調査した．ある新規故障状況に対する故障原因の推薦には，使用するデータ全てから作成した故障原因集合の各要素である，故障原因そのものを候補として使用している．

3.2 グラフ構築

本研究では，node2vec の学習に故障状況とその原因を関連させたグラフを入力とする．具体的には表 1 の形式になっている各学習データ 1 件ずつに対して，単語分割を行い，故障状況と原因の単語間での完全 2 部グラフを作成する．図 1 のように，各学習データの 2 部グラフを統合することで学習データ全体に関するグラフを作成する．本研究では，統合時に重複した故障状況と故障原因間の節点の組み合わせを数えたものをこの節点間の重みとしたため，最終的なグラフは故障状況と原因間の共起頻度の重み付きグラフとなる．

3.3 故障原因推薦手法

本研究では，新規の故障状況に対して，分散表現を元にして関連すると考えられる故障原因を推薦する．具体的には以下の順序で故障原因を推薦する．

1. 学習した分散表現を元に，新規の故障状況の分散表現を獲得する．
2. この故障状況の分散表現と学習データ内の各故障状況の分散表現のコサイン類似度を計算し，対応する故障原因の関連度とする．
   例えば表 1 の場合，新規故障状況と ‘冷却不足’ の類似度が 0.5 のとき，その新規故障原因と ‘クーラー氷結’ の関連度は 0.5 となる．
3. この関連度の高いものから順に，新規故障原因に対して関連した故障原因として推薦する．

本研究で学習した分散表現は節点単位であるが，この節点は故障状況，故障原因をそれぞれ分かち書きしたものであるので，表 1 のように故障状況は複数の節点から構成されている．そのため，本研究では故障状況内の各節点の分散表現を求め，それらを平均したものを故障状況の分散表現とした．

学習データにおいて複数の故障状況と関連する故障原因の関連度は最も高いものを使用している．関連度

表 2: 推薦実験での各手法における precision@k と mean average precision

	冷凍						火力	自販
	p@1	p@2	p@3	p@4	p@5	MAP	MAP	MAP
freq baseline	30.00	18.75	15.42	13.12	11.25	19.74	47.62	10.36
Skip-gram w/o pretrain	28.75	**24.38**	17.50	16.56	14.25	21.80	57.51	15.42
Skip-gram w/ pretrain	27.50	22.50	16.25	15.31	14.25	20.60	60.08	16.16
node2vec $p = 1, q = 0.25$	**31.25**	**24.38**	**18.33**	**16.88**	**14.50**	**23.74**	**65.06**	**17.45**

の値域は [-1, 1] である．また，同一の関連度の故障原因は学習データ中の頻度の高い順に並べた[3]．

4 障害レポートにおける故障状況の原因推定実験

4.1 実験設定

本研究では，冷凍庫に関する障害レポートである表 1 の形式のデータ 4,556 件を使用する．学習データとして 4,006 件，開発データとして 100 件，評価用データとして 450 件，ランダムに分割した．使用するデータ全てから作成した原因集合の要素数は 1,945 種であった．評価データでは一致した故障状況は統合している．そのため評価データの故障状況は 80 件になり，各故障状況に対する平均故障原因は 4.68 件となった．単語分割は MeCab[4]（IPADic 2.7.0）を使用した．

別分野のデータに対しても node2vec が有効か調査するために，火力発電と自販機に関する障害レポートに対してもそれぞれ実験を行った．火力発電は学習データが 660 件，評価データが 300 件であり，自販機は学習データが 830 件，評価データが 300 件である．推薦候補となる原因集合の要素数は火力発電が 74 種であり，自販機が 488 種であった．node2vec のハイパーパラメータは冷凍庫の分野で使用したものと同じものを使用する．

本研究は baseline として，新規の故障状況に関わらず，候補となる故障原因集合の各要素を学習データの頻度順で並べたもの（freq）を使用する．比較対象である単語分散表現は故障状況や故障原因を抜き出す前の障害レポートの部分を用いて学習している．さらに，この学習データのみでは単語分散表現の学習データとして不十分である可能性を考えて，Wikipedia 事前学習済みモデル[5]に対して学習データで再学習したモデルも用いて実験している．比較のため単語分散表現は Skip-gram モデル[6]を用いている．また node2vec のモデルは Grover and Leskovec の実装[7]を使用した．

Skip-gram，node2vec の次元数は 300 とした．node2vec のハイパーパラメータ p，q は開発データを用いて mean average precision（MAP）が最大になるように $p, q \in \{0.25, 0.5, 1, 2, 4\}$ から探索した[8]．

冷凍庫分野の評価は MAP とマイクロ平均で求めた precision@k（p@k）を用いた．火力発電，自販機分野の評価は MAP を用いた．

4.2 実験結果

各手法の障害レポートにおける故障状況の原因推定実験における precision@k と MAP スコアを表 2 に示す．baseline と比較すると分散表現を用いた方が性能が高いことがわかる．Skip-gram と node2vec を比較すると，同スコアまたは node2vec の方が優れていることがわかる．特に MAP では 1.94 ポイント向上していることがわかる．冷凍庫の分野では Skip-gram は事前学習モデルを用いるより，学習データのみを用いた場合の方が精度が高いことがわかる．

火力発電，自販機の分野でも node2vec を用いた推薦が MAP で最も高いスコアになっており，分野によらず node2vec の推薦手法が有効であることがわかる．

5 考察

本実験では，開発データを用いて node2vec のハイパーパラメータを $p = 1, q = 0.25$ とした．しかし，開発データに対して p，q を変えた時の一貫した MAP の偏りは発見できなかった．これは開発データのサイズが小さく，疎なデータであるためであると考えられる．

表 3 に冷凍庫の分野における実際の推薦例を示す．‘日配ケース蛍光灯不点灯’ という故障状況に対して，Skip-gram は ‘ＬＥＤ不点灯’ が最も類似した故障状況

[3]頻度も同じだった場合は辞書式に並べている．
[4]http://taku910.github.io/mecab
[5]https://github.com/Kyubyong/wordvectors
[6]https://github.com/RaRe-Technologies/gensim
[7]https://github.com/aditya-grover/node2vec
[8]この探索する値の集合は Grover and Leskovec と同一である．

表 3: 冷凍庫の分野でのある故障に対する各手法の p@1 の推薦例

	故障状況	故障原因
正解	日配ケース蛍光灯不点灯	**電子安定器不良**
Skip-gram w/o pretrain	ＬＥＤ不点灯	ＰＨＩＬＩＰＳ製ＬＥＤ（２５Ｗ）不良
node2vec $p = 1, q = 0.25$	蛍光管のちらつき	**電子安定器不良**

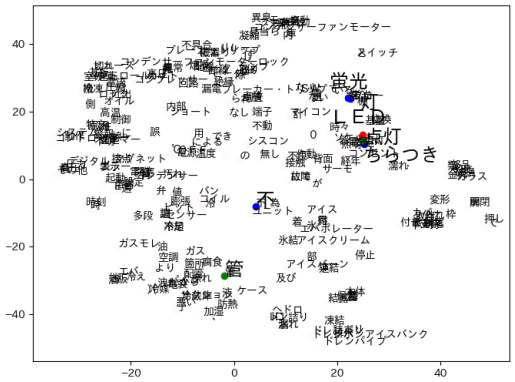

図 2: t-SNE を用いた node2vec の可視化

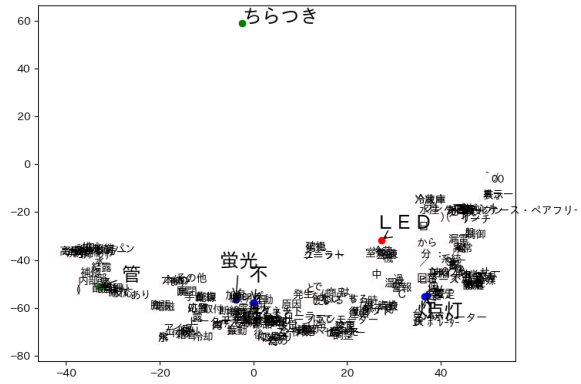

図 3: t-SNE を用いた Skip-gram w/o pretrain の可視化

であるとして対応した故障原因を推薦している．一方で node2vec は ‘蛍光管のちらつき’ が最も類似した故障状況だとして，対応した故障原因を推薦し，正解している．Skip-gram ではこの故障原因を 101 番目に推薦していることを確認した．表 3 内の故障状況を表す単語分散表現と節点分散表現を可視化[9]したものを図 2，3 に示す．図 2 から ‘ちらつき’ という単語が ‘蛍光’ や ‘点灯’ の近くにあることがわかり，node2vec ではこれらの単語は類似していると考えられる．学習データを確認したところ，故障状況である ‘ちらつき’ に対応した故障原因は，故障状況 ‘蛍光’ や ‘点灯’ と同様の故障原因であり，作成したグラフにおいてこれらは 2-Hop の関係であった．図 3 から Skip-gram では ‘ちらつき’ という単語が他の単語から遠くにあることがわかる．Skip-gram の ‘ちらつき’ の 10 近傍の単語を調べてみたところ，全ての単語が学習データに 1 回しか出現していなかった．一方で ‘蛍光’ や ‘点灯’ は学習データ中にそれぞれ 28 回，524 回出現している．Skip-gram は学習データの頻度に基づいた分布類似度を学習しており，そのため ‘蛍光’ や ‘点灯’ の遠くに存在していると考えられる．

6 おわりに

本研究では故障状況とその原因が結びついたデータが存在するとき，新規の故障状況に対してその原因を推薦する問題に取り組んだ．この問題設定において，故障状況と故障原因の関係性を考慮した node2vec の節点分散表現の方が単語分散表現よりも性能が高いことを示した．

障害レポートは，故障の状況と原因の他にどこの部品が故障したのか，という場所に関する情報も本来は含まれている．今回は問題設定を簡単にするため，故障した場所に関する問題設定は考慮していない．一方で，より現実的な設定を考えると故障場所も考慮した故障原因を推薦する手法及び評価を考える必要がある．

[9] 可視化には scikit-learn の t-SNE を用いた．頻度順に 250 単語と表 3 の単語を可視化している．

参考文献

[1] Hanyin Fang, Fei Wu, Zhou Zhao, Xinyu Duan, Yueting Zhuang, and Martin Ester. Community-based question answering via heterogeneous social network learning. In *Proc. of AAAI*, pp. 122–128, 2016.

[2] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proc. of KDD*, pp. 855–864, 2016.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013.

[4] Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, and Yueting Zhuang. Expert finding for community-based question answering via ranking metric network learning. In *Proc. of IJCAI*, pp. 3000–3006, 2016.