# ドメイン・ユーザー指向辞書の対話的構築

小比田 涼介　那須川 哲哉　吉田 一星　金山 博
日本アイ・ビー・エム株式会社　東京基礎研究所
kohi@ibm.com, {nasukawa, issei, hkana}@jp.ibm.com

## 1　はじめに

テキストマイニングの目的は，大量の生文書から意味のあるパターンを発見し，そのパターンの背後にある有益な知見を得ることである [5]．そして，その分析過程において不可欠となるのが単語辞書である [2]．例えば，車に関する文書から「どの車種がどういった機能を持っているのか」を知ろうとした時に，「車種に関する単語辞書（例 プリウス，マーチなど）」と「機能に関する単語辞書（例 自動アシスト，バックモニターなど）」を用意することによって，車種と機能の関係性を抽出できる．本稿では，こうした単語辞書を，機械による候補提示と人間からのフィードバックによって効率的に構築する手段について提案する．

上で述べた単語辞書は以下の二つの特徴を持つ．

(i) ドメインに依存し，一般知識では不十分である．

(ii) ユーザーの分析動機に応じて，一つの単語が異なる辞書に登録されうる．

例えば，「マーチ」という単語は「車名」の他に，「行進曲」や「都内私立大学」といった多義性があり，文書内での使われ方に依存して決められなければならない（i）．また，車名の「マーチ」であったとしても，「日産車」でもあり，「日本車」でもあり，「コンパクトカー」でもあるように，複数の観点から異なるまとめ方が可能である（ii）．つまり「ある単語が，どういった単語と共に，どのような辞書に登録されるか」は分析段階まで不明で，予め全てを用意するのは困難である．

テキストマイニングの現場において，こうした単語辞書はユーザーによって人手で構築され，作業負荷の高い項目となっている [8]．しかしながら，分析動機によって辞書が定義されるので，ユーザーの介入は不可欠である．そのため，現場からは，文書といくつかのフィードバックを元に，現在構築している辞書の性質を察知して，関連しうる単語候補を提示するようなシステムが求められている [8]．

さて，候補の選出に関しては，「各単語の特徴量ベクトルを定義し，そのベクトル空間での類似度が高いものを提示する」という手法が一般的であり，特徴量には Word2Vec など周辺単語分布に関するものが扱われ，似た文脈を持つ単語が候補として提示される [1, 3]．特徴量を当該文書内で構築することによって，単語のドメイン性を反映することはできるが，しかし，ユーザーが求めている辞書を推定するには不十分であり，フィードバックを候補選出に反映させる必要がある．

以上を踏まえ，本稿では対話型の辞書構築支援手法を提案する．提案手法はスコア計算（2.1 節）と重み学習（2.2 節）の二つに分かれる．スコア計算では，異なる観点から単語の類似性を捉える異なる特徴量を使ったモデルを独立に保持し，モデルごとのスコアの加重平均を取る．これは，多角的に類似性を捉え，多様な候補を提示すること，及び，重み調整によって，ユーザーの意向を反映することを意図している．重み学習は，上記の重みを調整する部分であり，提示候補に対するユーザーフィードバックから，現在どの特徴量を重視して候補を提示すべきかを更新する．学習方略としては，ユーザーが正解とした提示候補のうち，スコアが最小のものを最大化する．これによって，過学習を防ぎながら与えられた正解候補全体のスコアを上げ，未知の正解単語が候補として提示されるのを狙った．実験では，車の事故・故障に関する実在データで検証を行い，既存手法と比べ，提案手法が辞書構築の効率化に寄与することを確認した．

## 2　提案手法

本章では，類似性を多角的に捉え，かつ，ユーザーの意向を反映できるスコア計算とフィードバックからの重み学習について説明する．以降では「異なる特徴量関数を持つ独立モデル」のことを「モデル」と呼ぶ．

### 2.1　スコア計算

提案手法は，$|M|$ 個のモデルからなる集合 $M$，及び，各モデルに対応する重み $\boldsymbol{w} = (w_1, w_2, \cdots, w_{|M|})$

を持ち，単語 $a$ と単語 $b$ のスコアは，各モデル $M_i$ によるコサイン類似度の加重平均とする（式 1）．$f_i$ はモデル $M_i$ の特徴量関数で特徴量ベクトルを返す．$cos$ は二つのベクトルのコサイン類似度を算出する．$Z_i$ はモデル $M_i$ 内での z 値正規化を表しており，ただし，正規化後の負値は 0 に丸める．

$$score(a, b, \boldsymbol{w}) = \frac{\sum_i^M \boldsymbol{w}_i * Z_i(cos(f_i(a), f_i(b)))}{|M|} \quad (1)$$

異なる特徴量を持つモデルを独立に扱うことによって，各特徴量が**異なる観点から捉える類似性**も個別に扱うことができる．例えば,「マーチ」や「MARCH」といった同義語の場合は，n-gram のような固定的な文脈類似度が重要となり，一方で,「マーチ」「フィット」といったトピックが似ている単語を集める場合には，文脈を柔軟に捉えられる Word2Vec が役立つなどが考えられる．

加えて，モデルごとに重み付けを行うのは計算量における利点もある．それぞれのコサイン類似度は事前計算でき，かつ，学習する重みの数も高々モデル数であるため，対話場面における計算量を抑えられる．

### 2.2 重み学習

ユーザーが意図する単語を提示するために，フィードバックから適切な重み $\boldsymbol{w}$ を求める．ユーザーからのフィードバックは，シード単語 $s$，正解単語集合 $P$，誤り単語集合 $N$ を指す[1]．$s$ 及び $P$ の単語は，ユーザーが求める辞書に含まれる単語で，$N$ の単語は含まれない単語となる．例えば，ユーザーが車種名の辞書を構築する場合，$s$ には「マーチ」，$P$ には「フィット」「プリウス」，$N$ には「エンジン」「トヨタ」などが考えられる．そして，$\boldsymbol{w}$ の更新によって,「スイフト」や「フォレスタ」のスコアが高くなり，候補として提示されることが期待される．

提案する学習方略は，(A) シード単語 $s$ に対する正解単語集合 $P$ の最低スコアの最大化，(B) 誤り単語集合 $N$ のスコアの全体的な最小化，の 2 点からなる．

重み学習の疑似コードがアルゴリズム 1 である．まず，学習率 $\alpha$，更新回数 $K$，重み $\boldsymbol{w}$ を初期化する．学習率・更新回数はそれぞれ 0.1・50 と経験的に決め，重みは均等割りとした．各更新ステップでは，当該ステップにおける $P$ の中での最低スコア $p$ と $N$ からランダムに選んだ単語のスコア $n$ を得る．それぞれの勾配（$\frac{dp}{dw}, \frac{dn}{dw}$）を用いて，$p$ は上がる方向に，$n$ は下がる方向に学習率（$\alpha, \frac{\alpha}{2}$）を掛けて更新する．ただし，$\boldsymbol{w}$ の総和が 1 になるように，毎ステップ正規化を行う．

[1] $N$ は必須ではなく，与えられなかった場合は関連動作を全てスキップする．

**Algorithm 1** 学習アルゴリズム

INPUT : シード単語 $s$，正解単語集合 $P$，誤り単語集合 $N$
OUTPUT : 更新後の $\boldsymbol{w}$

SET $\alpha = 0.1, K = 50$
SET $\boldsymbol{w}$, s.t. $w_i = \frac{1}{|M|}$
**for** $i = 0$ to $K$ **do**
  $p = \min\{score(s, p_i, \boldsymbol{w}), p_i \in P\}$
  $n = score(s, random(N), \boldsymbol{w})$
  $\boldsymbol{w} \leftarrow \boldsymbol{w} + \frac{dp}{d\boldsymbol{w}} * \alpha - \frac{dn}{d\boldsymbol{w}} * \frac{\alpha}{2}$
  $w_i = \frac{w_i}{\sum \boldsymbol{w}}$
**end for**

## 3 実験

実在テキストに対して評価用のユーザー辞書を作成し，提案手法によって辞書に含まれている単語をどの程度候補として提示できるようになるかを検証する．

### 3.1 データ

実験データには，NHTSA[2]が公開している米国内での車の事故・故障に関する資料を使用する．このデータには，車種などのメタ情報の他に，具体的な故障内容に関する記述が含まれており，その部分を対象テキストとする．2016 年以降の 7,922 エントリーを対象とし，総文数は 51,347，総単語数は 945,877 である．このテキストに対して，実際の分析場面で想定しうる 6 つの名詞辞書（表 1）を作成し，評価に用いる．文及び単語の分割，品詞タグ付け，係り受け解析は全て UDpipe[6][3]で行う．また，提示候補は品詞タグの過半数が「NOUN（名詞）」あるいは「PROPN（固有名詞）」と推定された単語で予め絞る．

表 1: 評価辞書一覧

| 辞書 | 内容 | 数 * | 例 |
|---|---|---|---|
| Engine | エンジン関連 | 27 | engine, gasoline |
| Exterior | 外装関連 | 32 | light, bumper |
| Interior | 内装関連 | 62 | audio, handle |
| Driver | 運転席関連 | 28 | handle, horn |
| Carname | 車種名 | 79 | prius, tundra |
| Company | 会社名 | 58 | toyota, bmw |

* 数はデータの全文を使った場合のもの．データを分割した際は，分割データの中で頻度が 5 以上の単語を辞書エントリーとして使用．

[2] https://www.nhtsa.gov/（2018 年 9 月 27 日ダウンロード）
[3] http://hdl.handle.net/11234/1-1659

表 2: 比較手法

| 手法 | 概略 |
|---|---|
| mix | 加重平均 |
| mix+grad | 加重平均 + 重み学習（正例） |
| mix+gradneg | 加重平均 + 重み学習（正例・負例） |
| ng | n-gram |
| head | head-dependency |
| w2v | Word2Vec |
| svm | SVM で重み学習 + スコアリング |
| svmw | SVM で重み学習 + 加重平均） |

### 3.2 モデル

本実験では，n-gram･BoW･PMI･head-dependecy･child-dependency･Word2Vec[4]の 6 つの特徴量を用いる．2.1 節で述べたように，単語間のスコアは，これら 6 つの独立した特徴量別モデルから得られる，6 つのコサイン類似度を加重平均したものである．

比較手法として表 2 の 8 つを取り上げる．提案手法は，mix･mix+grad･mix+gradneg の 3 つであり，重み学習の有無において異なる．単一の特徴量のみを使うモデルが ng・head・w2v で，それぞれの特徴量のみによるコサイン類似度で候補を提示する[5]．加えて，分類問題として学習するモデルに svm･svmw を挙げ，こちらでは，mix と同じ 6 つのモデルによるコサイン類似度を入力とする 2 値分類器を学習する．svm・svmw 共に，与えられた正例・負例から重みを学習する点は共通しているが，その使用方法が異なる．svm はスコアリングも分類器の推定確率値を流用するが，svmw では学習した重みを提案手法のスコアリング関数で用いる．

### 3.3 評価

本実験では 2 種のシミュレーションで評価する．ここで，候補に対して，辞書を参照して正解候補 $P$ と誤り候補 $N$ に機械的に振ることを「アノテーション」と呼ぶ．

**(I)** シード単語に対する初期候補（10･20･30 個）にアノテーションを施した後，次の 30 個の候補中にいくつ正解候補が含まれているかを評価する．学習なしの手法（mix, ng, head, w2v）はそのまま次の候補を提示し，学習ありの手法（mix+grad, mix+gradneg, svm, svmw）は初期候補へのアノテーション結果から学習した後に，次の候補を提示する．

[4]n-gram は 1 ～ 5 まで，BoW・PMI はウィンドウ幅 5 で定義した．n-gram・ BoW・head-dependency・child-dependency はカウントベースのスパースな特徴量である．

[5]BoW・PMI・child-dependency についても実験を行ったが，全体の結論に影響がないので省略する．

**(II)** シード単語に対して候補を 10 個ずつ提示し，100 個の単語にアノテーションを施す．各ステップにおいて，辞書単語の何割をカバー出来ているかを評価する．学習ありのモデルはステップごとに学習を行う．

### 3.4 結果

評価 (I) の結果が図 1 である．提案手法はいずれの条件でも単一特徴量，および，SVM よりも多くの正解単語を提示できている．ただし，重み学習の効果はアノテーション数が 10 の時は見られず（図 1 左），20 以上のアノテーションを与えることで学習によって候補中の正解単語数が増える（図 1 中右）．また，特にデータ量が少ない時にその効果が顕著である（< 20000 文）．評価 (II) の結果が図 2 で，同図左がデータが小さい場合，同図右がデータが大きい場合である（5000 文 vs. 50000 文）．いずれの場面でも提案手法のカバー率が高く推移している．データが小さい場合は，3 ステップ目（i.e. 30 単語にアノテーションを施した時点）から，重み学習モデル（mix+grad, mix+gradneg）が学習なしのモデル（mix）を上回っている．データが大きくなると学習効果は薄れるが，ステップが進むにつれて，アノテーション数が増え，学習によってより多くの正解単語をカバーできるようになっている．データが小さい時は SVM もそれなりのカバー率に達しているが，データが大きくなるとステップ後半での伸びが悪くなる．おそらく辞書の一部の代表的な単語にオーバーフィットしてしまい，マイナーな単語をうまく提示できないためと考えられる．

## 4 分析

本手法は，ユーザーフィードバックに応じて同じ単語に対しても異なる候補を提示することが期待される．ここでは全文を用いた場合の mix+gradneg モデルの振る舞いを分析する．シードとして「TOYOTA」を与えると，初期候補として表 3 の左列が得られる．「日本車」辞書を想定して「Honda」「SUBARU」「 LEXUS」を正例として与えた場合の候補が表 3 中列で，「日本車＋米国車」辞書を想定して，さらに「Ford」「Chevrolet」「Chrysler」「GMC」を正例として与えた場合の次の候補が表 3 右列である．日本車のみを与えた場合は，「Acura」や「Mitsubishi」といった新たな日本車メーカーが提示されるのに対して，米国車も与えると「Dealer」「Dealership」といった車メーカーに関する一般的な単語が提示される．このようにユーザーの

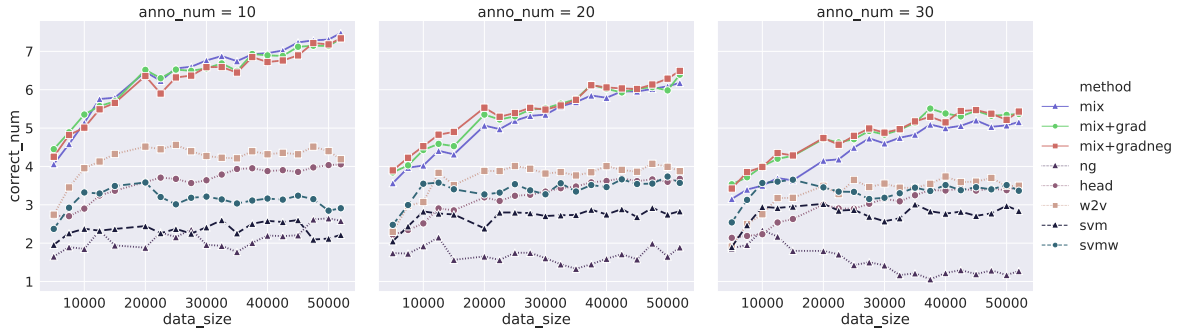


図 1: 学習後の平均正解単語数．Y 軸が平均正解単語提示数，X 軸が使用文数，ファセットがアノテーション数．

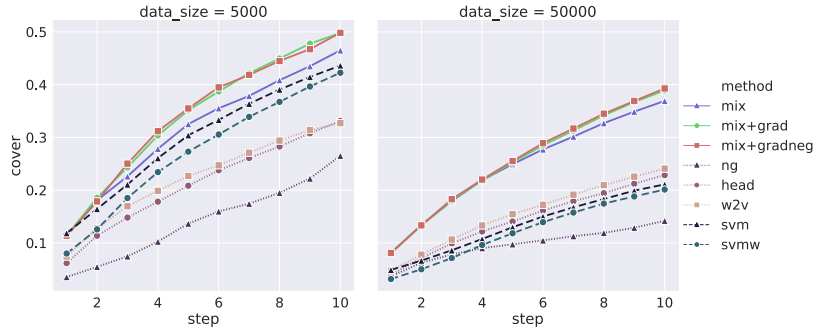


図 2: ステップごとの正解単語カバー率

表 3: 「TOYOTA」に対する学習前・学習後候補

| 初期候補 | 日本車で学習後 | 日本・米国車で学習後 |
|---|---|---|
| **Honda** | Audi | Dealer |
| *Ford* | *Cadilac* | Audi |
| KIA | Jayco | VW |
| *Chevrolet* | Hyundbe | *Cadilac* |
| Hyundai | VW | *Jeep* |
| Volkswagen | *Jeep* | Dealership |
| **SUBARU** | Volvo | *GM* |
| **LEXUS** | **Acura** | Volvo |
| *Chrysler* | **Mitsubishi** | Hyundbe |
| *GMC* | *Lincorn* | Mercedes |

(左) 初期候補，(中) 日本メーカーで学習後の候補，(右) 日本・米国メーカーで学習後の候補．
太字は日本車、斜体は米国車.

フィードバックに反応して，動的により適した候補を返すことができている．

## 5 おわりに

本稿ではテキストマイニングにおける辞書構築の課題を提示し，その解決策として，対話型の辞書構築支援手法を提案した．辞書の半自動・自動構築は，同義語や上位語・下位語といった一般的単語性質としての試みはなされてきているが [3]，今回のようにドメインやユーザーに依存して定義される環境下での手法はほとんど議論されていない．どのように既存の技術を適用できるのか，あるいは，できないのかを評価データの整備と共に行っていくことが課題である．

提案手法に関連する研究分野としては，課題に応じて適切な特徴量を選択することを指す Feature Selection[7] や Feature Fusion[4] が挙げられる．本稿では，辞書構築という目的・場面における特徴量設計，及び，その選択方略を議論した．

## 参考文献

[1] Desheng Cai, Jingjing He, Gong-Qing Wu, and Xuegang Hu. Synonymous entity recognition based on feature fusion. *IEEE International Conference on Big Knowledge*, 2017.

[2] Shantanu Godbole, Indrajit Bhattacharya, Ajay Gupta, and Ashish Verma. Building re-usable dictionary repositories for real-world text mining. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, 2010.

[3] Dishan Gupta, Jaime Carbonell, Anatole Gershman, Steve Klein, and David Miller. Unsupervised phrasal near-synonym generation from text corpora. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[4] Utthara Gosa Mangai, Suranjana Samanta, Sukhendu Das, and Pinaki Roy Chowdhury. A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical Review*, Vol. 27, No. 4, 2010.

[5] T. Nasukawa and T. Nagano. Text analysis and knowledge mining system. *IBM Syst. J.*, Vol. 40, No. 4, October 2001.

[6] Milan Straka and Jana Straková. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 2017.

[7] B. Xue, M. Zhang, W. N. Browne, and X. Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, Vol. 20, No. 4, 2016.

[8] 哲哉那須川. テキストマイニングを使う技術/作る技術 : 基礎技術と適用事例から導く本質と活用法. NetLibrary, 2009.