指数型分布族埋め込みで可視化する レコード・チャイナの言説

呉 東文
（ディップ株式会社商品開発本部データサイエンティスト）
2023 年 12 月 25 日

*本論考は計量政治学とデータを用いた分析で導き出された筆者の個人の意見であって、所属する組織の見解を示すものではありません。

要 旨

本稿では、ベイズ機械学習の手法である指数型族埋め込みを利用して、レコード・チャイナ（Record China）から出版される記事の中にどのような言説が含まれているかを分析する。具体的には、指数型族埋め込みを利用してレコード・チャイナ記事の中の単語の意味を推定する。推定の結果、中国が自国の民主主義が優れているという主張と、アメリカこそが脅威になっているという言説が定量的に確認できた。今後データ量が拡張されれば、レコード・チャイナの言説の変化やレコード・チャイナの言説と日本の一般的なメディアの言説の違いなどの可視化も可能になるだろう。（論稿の内容はあくまでも個人の見解であり、著者が所属するディップ株式会社とは一切関係ない。）

はじめに

本稿では、一橋大学の市原麻衣子教授から提供を受けた、スクレイピングデータである 2021 年と 2022 年のレコード・チャイナ（Record China）の全記事テキストデータを利用する。そして Ichihara(2020)が指摘するようにレコード・チャイナのサイト運営に影響を与えていると思われる中国が、日本で広めようとしている言説の背後にある思想をベイズ機械学習で可視化する。

具体的にはルドルフら（Rudolph et al. 2016）が提案した指数型分布族埋め込み（exponential family embeddings）で単語同士の同時出現関係を解析する。例えば、レコード・チャイナの記事の中で「民主」と似たような文脈で現れる単語や、「アメリカ」と似たような文脈で現れる単語を抽出することで、レコード・チャイナは「民主」と「アメリカ」をそれぞれどのように論じているかを可視化することができる。

本稿は筆者が以前に行ったレコード・チャイナ分析（呉 2023）と同じデータセットを利用しているため、基礎的なデータの可視化については省略する。また、前回の論考と同様本稿も、レコード・チャイナから出版されている記事のうち、レコード・チャイナ記者が執筆した記事、あるいは人民網日本語版から転載された記事に絞って分析する。

分析手法

ここではまず、指数型分布族埋め込みが属する埋め込み系の手法の初期の代表例である word2vec の概念を説明する。次に本稿の分析で利用する指数型分布族埋め込みを説明する。ただし、数学的記述を避けるため、以下の説明は主に概念の説明に留める。

word2vec 系の手法の概念と用途

簡潔に説明すると、word2vec とはもともとコンピューターサイエンスの世界で提案された、単語をベクトル化してその言葉の意味を精緻にモデリングする手法である。ミコロフら（Mikolov et al. 2013a,b）などが初期の論文として有名である。ベクトル化とは、計量政治学や計量経済学でいう隠れ変数（latent variable）の設定と似た概念で、一つの単語の背後に複数個のパラメータが存在するという仮定である。隠れ変数で単語の意味を推定する方法については、複数の方法があるが、本稿でも採用するネガティブサンプリング（negative sampling）を中心に説明する。

ネガティブサンプリングとは、本物のデータの中に偽のものを混ぜて、モデルに渡されたものは本物なのか偽物なのかを学習させることで、言語の知識を教える概念である。

具体的な例として、分割処理をした実際のレコード・チャイナの文を以下に提示する。

新年 中国 （？？？）[1] 中国人 自信

[1] 本稿では、（？？？）に入る単語を目標単語と呼ぶ。

次に、(？？？)に二つの単語を挿入する。どちらが本物かを検討せよ。

新年 中国（台頭）中国人 自信
新年 中国（フィリーチーズステーキ）中国人 自信

あなたは瞬時に前者が本物のレコード・チャイナの記事の内容で、後者は筆者が適当に挿入した単語だと判断できただろう。しかし、なぜあなたは「新年」、「中国」、「中国人」、「自信」に囲まれた単語として「台頭」がふさわしく、「フィリーチーズステーキ」はふさわしくないと判断できるのだろうか。それは、「新年」、「中国」、「中国人」、「自信」という文脈がこの文章についての知識を提供しているからである。より具体的には、おそらく「中国」と「自信」が特に答えの判定に有用であろう。中国の自信に関する言説なので「台頭」が入るのはもっともらしく、アメリカ料理の名前が現れるのは唐突すぎるのである。

このように、単語の前後に存在する単語（文脈）から単語の出現を予測することで、単語の意味をモデルに理解させる手法が、word2vec なのである。そして、ミコロフら（Mikolov et al. 2013b）などの複数の論文で指摘されているように、推定されたベクトルは単語の意味を保持しており、単語の類似度測定で利用できることが知られている。なお、政治学での word2vec 系の手法の応用に関する主な参考文献としてはロドリゲスら（Rodriguez et al. 2022）が挙げられる。

指数族埋め込み

指数型分布族埋め込みは、word2vec から派生したベイズ機械学習の手法である。指数型分布族埋め込みは、まず埋め込みベクトル（embedding vector）と文脈ベクトル（context vector）という二種類のベクトルを定義する。そして、目標単語の埋め込みベクトルとその文脈にある単語の文脈ベクトルの和の内積をとり、これを η とする。

$$\eta_{台頭} = embeddingベクトル_{台頭}{}' * \left(contextベクトル_{新年} + contextベクトル_{中国} + contextベクトル_{中国人} + contextベクトル_{自信}\right)$$

$$\eta_{フィリーチーズステーキ} = embeddingベクトル_{フィリーチーズステーキ}{}' * \left(contextベクトル_{新年} + contextベクトル_{中国} + contextベクトル_{中国人} + contextベクトル_{自信}\right)$$

そしてηをベルヌーイ分布のパラメータとして、

$$本物なのかデタラメなのかフラグ \sim ベルヌーイ分布(\eta_{台頭})$$

$$本物なのかデタラメなのかフラグ \sim ベルヌーイ分布(\eta_{フィリーチーズステーキ})$$

をサンプリングさせる。モデルの性能が良ければ、前者のフラグに 1（本物）が、後者のフラグに 0（デタラメ）がサンプリングされるはずである。

指数型分布族埋め込みの最大のメリットは、ベイズ統計学・ベイズ機械学習の他の手法と柔軟に組み合わせられるところにある。例えば、埋め込みベクトルを所属政党別で変動させれば、政党に

よる言葉の意味の違いを緻密に守ることができる(Rudolph et al. 2017)。また、埋め込みベクトルを時間と共に変動させれば、言葉の意味の変化を可視化できる(Rudolph and Blei 2018)。

本稿ではデータの少なさを理由に、埋め込みベクトルに上述のような階層ベイズ的な構造を持たせていない。しかし、今後データが増えた場合、モデルのスタンコード(Stan code)を少し変化させることでレコード・チャイナの言説の時系列的な変化や、レコード・チャイナと一般的な日本のメディアの言説の違いを可視化できる。

データ分析

前処理

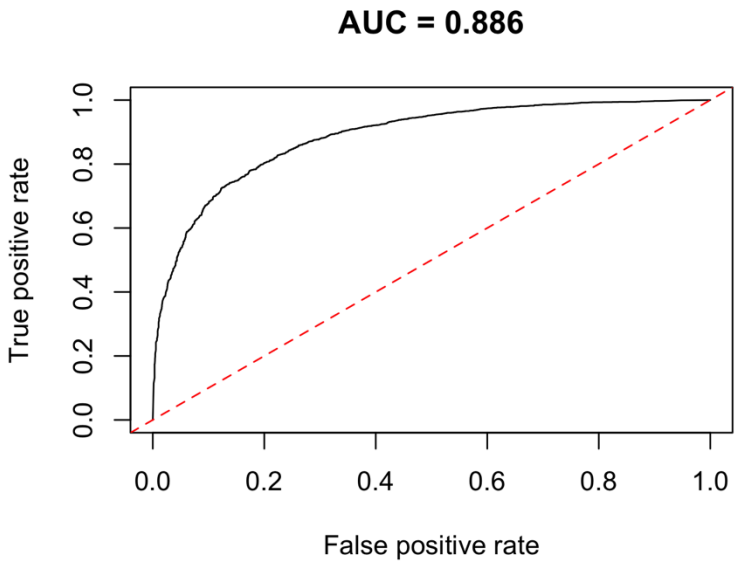
前回の論考(呉 2023)と大きく変わらないものの、本稿の分析でも、市原麻衣子教授から提供を受けたデータの中で、出典がレコード・チャイナか人民網日本語版に該当する記事のみを対象とする。日本語の処理に関しては、ひらがな、カタカナ、漢字、ローマ字以外のものは全てスペースに置き換える[2]。次に、RMeCab 経由で mecab を利用して名詞だけを残した。word2vec 系の手法の場合、単語をより多く残すことが多いが、限られたローカルマシーンのスペックと執筆時間のため、やむをえず前回の記事とほぼ同量の単語を消去した。また、全ての単語を id 化し、その周辺の 3 つの単語を文脈とする。さらに、上述のように文脈を固定し、目標単語の id だけをシャッフルさせたデタラメのデータも含めたデータフレームを作成した。最後に、データフレームに入っている 5,000 件のデータを検証データとして手元に残し、他のデータを筆者がスタンで作成した指数型分布族埋め込みベイズ機械学習モデルに投入する。埋め込みベクトルと文脈ベクトルの次元数は 50 に設定した。今回の分析は既存のパッケージを利用していないため、透明性のため Stan のコードは付録で確認できる。

精度確認

まず、モデルに渡していない検証データについて、果たしてモデルは単語の真偽を判別できるかを確認する。精度確認の指標としては、受信者操作特性(Receiver Operating Characteristic: ROC)曲線の曲線下の面積(Area Under the Curve: AUC)を利用する。ROC 曲線は簡単にいうと、45 度線から離れれば離れるほど精度が良く、45 度線と重なればランダムな分類に等しいことを示す。

[2] 正規表現でいうと[^ー-龠ぁ-んーァ-ケーa-zA-Z]に該当するものは全てスペースに置き換えた。

図 1 単語の真偽判定タスクにおける AUC



出典：著者作成

図 1 からわかるように、本モデルの検証データについて、単語の真偽判定タスクにおける AUC は 0.8 を超えており、一般的には非常に精度が良いといえる。したがって、モデルはある程度レコード・チャイナの単語出現パターン・言説パターンを学習していると判断しても良いであろう。

コサイン類似度から見た言説

コサイン類似度とは、単語ベクトルの類似度を測る指標である。本指標を用いれば、単語が利用される文脈や意味の近さを判断できる。

本分析はレコード・チャイナのデータのみ利用しており、いわゆる「学習済みモデル」等を使用していないため、モデルが推定したベクトル同士の類似度は、純粋にモデルの定式化と前処理をもとに得られたレコード・チャイナの言説を意味する。

まず、簡単なものから確認すると、レコード・チャイナにおいて「道路」と最もコサイン類似度が高い 10 単語は下記の通りである。

表1 レコード・チャイナにおいて「道路」と最もコサイン類似度が高い 10 単語

単語	コサイン類似度
道路	1.0000000
鉄道	0.7952032
区間	0.6528857
駅	0.6473283
キロメートル	0.6423741
大橋	0.6331041
車両	0.6313493
列車	0.6180682
橋	0.6118205
走行	0.6033028

「鉄道」、「駅」、「車両」など交通関連の単語が現れている。

次に、「空港」とコサイン類似度の高い単語はやはり交通系の単語であるが、より公共交通に関連する単語が現れ、「道路」とコサイン類似度の高い「大橋」や「車両」などの単語は見られない。

表2 レコード・チャイナにおいて「空港」と最もコサイン類似度が高い 10 単語

単語	コサイン類似度
空港	1.0000000
駅	0.6233985
到着	0.5561386
航空	0.5314159
鉄道	0.5304486
地下鉄	0.5278712
郵便	0.5166218
乗客	0.5165372
便	0.5122190
飛行機	0.5120097

さらに「川」とコサイン類似度の高い単語を確認すると、「池」、「山」、「魚」、「湖」などの自然に関する単語が現れた。

表3 レコード・チャイナにおいて「川」と最もコサイン類似度が高い 10 単語

単語	コサイン類似度
川	1.0000000
池	0.6394781
大橋	0.6306532
黄河	0.5989985
山	0.5945901
海水	0.5860707
峡	0.5855116
魚	0.5777902
流域	0.5767979
湖	0.5764602

最後に「ネット」とコサイン類似度の高い単語を確認する。

表4 レコード・チャイナにおいて「ネット」と最もコサイン類似度が高い 10 単語

単語	コサイン類似度
ネット	1.0000000
インターネット	0.6724772
書き込み	0.5358616
動画	0.5171089
削除	0.4966197
コメント	0.4766648
投稿	0.4734309
コンテンツ	0.4703328
事実	0.4548065
クラウドコンピューティング	0.4545467

「インターネット」、「書き込み」、「動画」、「コメント」などネット関連の単語が出現していることが確認できる。

このように、ROC 曲線の AUC でも、いくつかの単語をピックアップして確認する結果でも、本モデルはレコード・チャイナでの単語の利用パターンを推定できていることを示している。

では、ここからは法学・政治学的に意味ある単語の状況を確認する。まず「民主」とコサイン類似

度の高い単語を確認すると驚くべきことに中共と中国共産党が上位に出てきた。

表5 レコード・チャイナにおいて「民主」と最もコサイン類似度が高い 10 単語

単語	コサイン類似度
民主	1.0000000
中共	0.7087134
政治	0.6484074
民族	0.6299390
独立	0.6253643
野党	0.5998672
選挙	0.5954480
中国共産党	0.5883779
軍国	0.5872631
党	0.5770189

これは但見(2022)が指摘したように、中国が「美国(米国)式民主よりずっと良い」、すなわち、より素晴らしい「中国的民主」を海外に宣伝していることが定量的にレコード・チャイナを通しても確認できたといえよう。これは、サイト運営者が日本でも同様の言説を広めようとしていることを示唆する。

次に、「アメリカ」とコサイン類似度が高い単語を確認すると、上位には「アメリカ」とヨーロッパ関連の国名・地域名が現れたが、「スパイ」と「脅威」も「アメリカ」と似たような文脈で使用されている。

表6 レコード・チャイナにおいて「アメリカ」と最もコサイン類似度が高い 10 単語

単語	コサイン類似度
アメリカ	1.0000000
米国	0.5713009
欧米	0.5136229
ロシア	0.4954124
ドイツ	0.4741628
米	0.4739354
スパイ	0.4701061
欧州	0.4692116
ン	0.4647487
脅威	0.4597416

指数型分布族埋め込みモデルによる、モデル推定の理由を判断することは難しいが、「アメリカ」と「脅威」が同時に使用されているレコード・チャイナの記事を確認すると、次のように記述されている。

> 「米政府系メディアのボイス・オブ・アメリカは 27 日、代表的な SNS のフェイスブックとツイッターが、親米的な内容を投稿する偽アカウントを排除したと伝えた。排除されたアカウントは西側の立場に基づきロシアや中国、イランに対する非難を投稿していた。スタンフォード大学とSNS分析会社のグラフィカのリポートによると、これらの偽アカウントはフェイスブックとツイッターのサービス規約に違反し、欺瞞(ぎまん)的手段を利用して中東と中央アジアに向けて西側寄りの情報発信をしていた。」(Record China 2022a)

つまり、中国がアメリカからの「フェイクニュース」の「被害者」、つまりアメリカが「脅威」になっていることが主張されている。さらに、中国による、アメリカを悪者と主張する言説は、例えばこの記事でも確認できる。

最後に、「台湾」とコサイン類似度の高い 10 単語を確認する。

表6 レコード・チャイナにおいて「台湾」と最もコサイン類似度が高い 10 単語

単語	コサイン類似度
台湾	1.0000000
本土	0.7017824
海峡	0.5893464
パイナップル	0.5561649
香港	0.5517208
尖閣諸島	0.5513686
武力	0.5498675
大陸	0.5427460
靖国神社	0.5193581
反発	0.5146651

まずコサイン類似度が一番高いのは「本土」であり、これは「台湾は中国の一部である」と主張する「一つの中国」言説を広めようとしていると見なされうる。

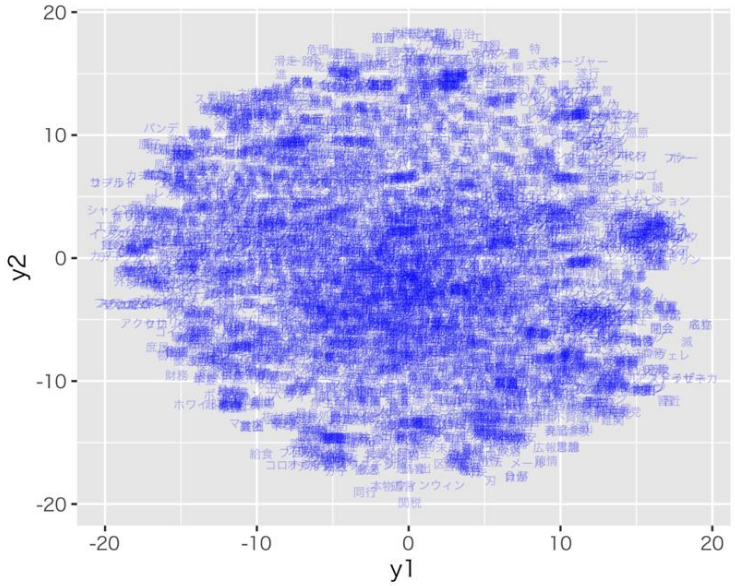
T-SNE を利用した可視化

最後に、ベクトルを二次元空間に映して可視化する手法である T 分布型確率的近傍埋め込み法（T-distributed Stochastic Neighbor Embedding: T-SNE）を利用して（Van der Maaten et al. 2008）、指数型分布族埋め込みモデルの推定結果をより視覚的に把握する。

もちろん、T-SNE はコサイン類似度とは別の概念のため、コサイン類似度が高いからといって、T-SNE で可視化する際の距離も近くなるわけではない。それでも、人間は視覚で様々な判断を行うため、全体の推定結果を視覚的に把握できれば、深掘りすべき点に関する仮説が出てくることが多い。

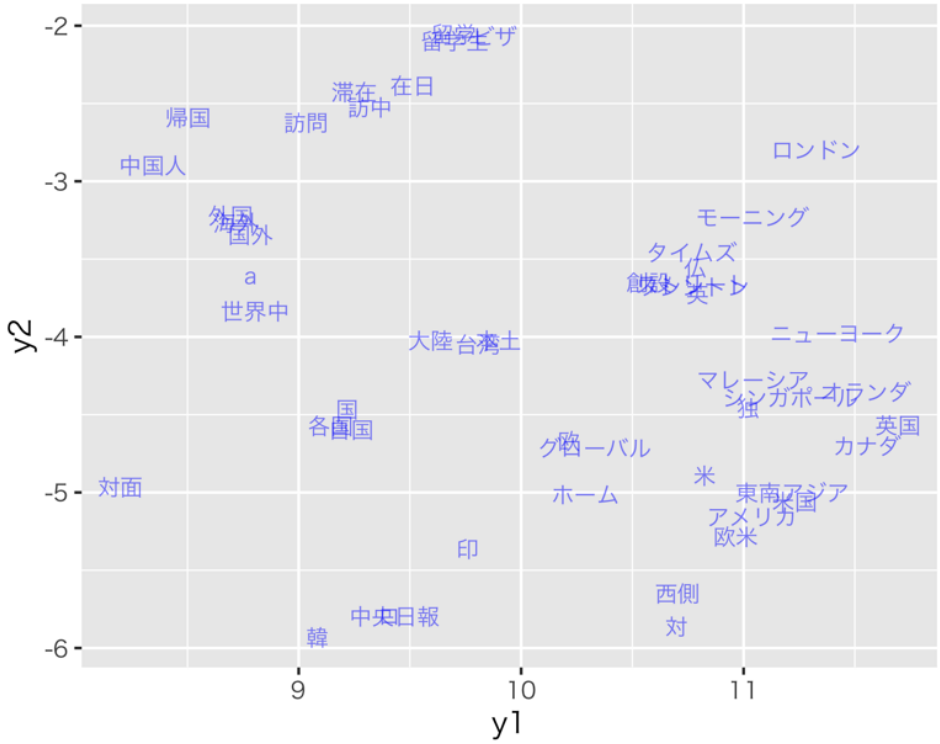
まず、埋め込みベクトル全体を可視化したものが図 2 である。さらに「台湾」付近にズームインすると、図３のように「大陸」と「本土」の距離が近いことが確認できる。前述のコサイン類似度を利用した分析においても指摘したが、これは、「台湾は中国の一部である」とする「一つの中国」を主張する言説を広めている分析結果を補強する。

図２ 指数型分布族埋め込みにおける埋め込みベクトルの T-SNE



出典:著者作成

図３「台湾」付近の T-SNE



出典:著者作成

結論

本稿では、word2vec のベイズ版の派生型である指数型分布族埋め込みを利用して、レコード・チャイナが拡散する言説を、言葉の意味の観点から分析した。さらに中国が、西欧的な民主主義と比較して「中国的民主」を称賛する言説とアメリカが脅威であるとする言説を広めようとしていることを可視化した。

これは、構造トピックモデルを利用した分析では捉えられなかった傾向であり、テキスト・アズ・データ（text as data）の分析を行う際には、さまざまな手法を併用することでより深い示唆を抽出することができる。

ただし、前回の論考（呉 2023）と同様、本稿においてもデータ量の制限によって、指数型分布族埋め込みの強みである階層ベイズ的な構造を生かすことができていない。今後取得できるデータが増えた場合、モデルの構造をより複雑にし、言説の変化や言説の相異など、より政治学的に意義のある側面に焦点を当てた更に緻密な分析が期待される。

参考文献

Ichihara, Maiko (2020). "Is Japan Immune from China's Media Influence Operations?" *The Diplomat*. (https://thediplomat.com/2020/12/is-japan-immune-from-chinas-media-influence-operations/ accessed 2023-11-24)

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). "Efficient Estimation of Word Representations in Vector Space." *arXiv preprint arXiv,* 1301, 3781.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean (2013b). "Distributed Representations of Words and Phrases and Their Compositionality." *Advances in Neural Information Processing Systems,* 26.

Record China (2022a).「フェイスブックとツイッターが「親米的偽アカウント」を排除」(https://www.recordchina.co.jp/b900239-s25-c100-d0198.html 2023 年 12 月 10 日最終閲覧)

Rodriguez, Pedro L., and Arthur Spirling (2022). "Word Embeddings: What Works, What Doesn't, and How to Tell the Difference for Applied Research." *The Journal of Politics,* 84(1), 101-115.

Rudolph, Maja, Francisco Ruiz, Stephan Mandt, and David Blei (2016). "Exponential Family Embeddings." *Advances in Neural Information Processing Systems,* 29.

Rudolph, Maja, Francisco Ruiz, Susan Athey, and David Blei (2017). "Structured Embedding Models for Grouped Data." *Advances in neural information processing systems,* 30.

Rudolph, Maja, and David Blei (2018). "Dynamic Embeddings for Language Evolution." *Proceedings of the 2018 wWorld Wide Web Conference*, 1003-011.

Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing Data Using t-SNE." *Journal of machine learning research* 9.11 (2008).

但見亮「中国的民主の挑戦(1)－「普遍的価値」をめぐって一」『一橋法学』21(2)、2022 年、165-199 頁。

Appendix

Stan のコード

```
  functions {
  real partial_sum_lpmf(
    array[] int result,
    int start, int end,

    array[] int word,

    array[] int word_before_1, array[] int word_after_1,
    array[] int word_before_2, array[] int word_after_2,
    array[] int word_before_3, array[] int word_after_3,

    array[] vector word_embedding,
    array[] vector word_context
  ){
    vector[end - start + 1] lambda;
    int count = 1;
    for (i in start:end){
      // Rudolph et al.(2016)の式(2)
      // 目標単語の embedding とその文脈(context)にある単語の context vector の和の内積を取る
      lambda[count] = word_embedding[word[i]] ' *
      (
        word_context[word_before_1[i]] + word_context[word_after_1[i]] +
        word_context[word_before_2[i]] + word_context[word_after_2[i]] +
        word_context[word_before_3[i]] + word_context[word_after_3[i]]
      );
      count += 1;
    }
    return (
      // Rudolph et al.(2016)の式(1)のように本物かデタラメかを示すフラグをサンプリング
      bernoulli_logit_lupmf(result | lambda)
    );
  }
}
data {
  int<lower=1> N; //学習データ数
```

```
  int<lower=1> K; //embedding 次元数
  int<lower=1> word_type; //単語

  //学習データ
  array[N] int<lower=1,upper=word_type> word; // 目標単語の id
  array[N] int<lower=1,upper=word_type> word_before_1; // 目標単語の 1 個前の単語の id
  array[N] int<lower=1,upper=word_type> word_after_1; // 目標単語の 1 個後の単語の id
  array[N] int<lower=1,upper=word_type> word_before_2; // 目標単語の 2 個前の単語の id
  array[N] int<lower=1,upper=word_type> word_after_2; // 目標単語の 2 個後の単語の id
  array[N] int<lower=1,upper=word_type> word_before_3; // 目標単語の 3 個前の単語の id
  array[N] int<lower=1,upper=word_type> word_after_3; // 目標単語の 3 個後の単語の id
  array[N] int<lower=0,upper=1> result; // 本物かデタラメかを示すフラグ。0:デタラメ、1:本物

  int<lower=0> val_N; //検証データ数
  //検証データ
  array[val_N] int<lower=1,upper=word_type> val_word; // 目標単語の id
  array[val_N] int<lower=1,upper=word_type> val_word_before_1; // 目標単語の 1 個前の単語の id
  array[val_N] int<lower=1,upper=word_type> val_word_after_1; // 目標単語の 1 個後の単語の id
  array[val_N] int<lower=1,upper=word_type> val_word_before_2; // 目標単語の 2 個前の単語の id
  array[val_N] int<lower=1,upper=word_type> val_word_after_2; // 目標単語の 2 個後の単語の id
  array[val_N] int<lower=1,upper=word_type> val_word_before_3; // 目標単語の 3 個前の単語の id
  array[val_N] int<lower=1,upper=word_type> val_word_after_3; // 目標単語の 3 個後の単語の id
}
parameters {
  array[word_type] vector[K] word_embedding; // embedding
  array[word_type] vector[K] word_context; // context vector
}
model {
  // embedding と context vector のサンプリング(標準正規分布)
  for (p in 1:word_type){
    target += normal_lupdf(word_embedding[p] | 0, 1);
    target += normal_lupdf(word_context[p] | 0, 1);
  }

  int grainsize = 1;

  // Reduce sum
  target += reduce_sum(
```

```
    partial_sum_lupmf, result, grainsize,
    word,
    word_before_1, word_after_1,
    word_before_2, word_after_2,
    word_before_3, word_after_3,
    word_embedding,
    word_context
  );
}
generated quantities {
  // 検証データに対する分類結果生成
  array[val_N] int predicted;
  for (n in 1:val_N){
    predicted[n] = bernoulli_logit_rng(
      word_embedding[val_word[n]] ' *
      (
        word_context[val_word_before_1[n]] + word_context[val_word_after_1[n]] +
        word_context[val_word_before_2[n]] + word_context[val_word_after_2[n]] +
        word_context[val_word_before_3[n]] + word_context[val_word_after_3[n]]
      )
    );
  }
}
```

呉東文プロフィール
ディップ株式会社商品開発本部データサイエンティスト。国際・行政修士（専門職）。一橋大学国際・公共政策教育部国際・公共政策専攻（国際・行政コース）専門職学位課程修了後、東京で民間のデータサイエンティストとして就職し、現在に至る。専門はテキストアズデータ、因果推論、大規模ベイズ機械学習をはじめとする計量政治学と計量経済学。ブログは https://qiita.com/Gotoubun taiwan。