Text Mining Maniax [word2vec編]

RとRMeCabによる
日本語の単語埋め込みの基礎

後藤和智
（後藤和智事務所 OffLine）

Text Mining Maniax

[word2vec 編]

R と RMeCab による
日本語の単語埋め込みの基礎

著：後藤和智（後藤和智事務所 OffLine）
発行：2023 年 8 月 13 日
(コミックマーケット 102)

Text Mining Maniax [word2vec 編]
R と RMeCab による日本語の単語埋め込みの基礎

はじめに

91 冊目の同人誌となります、後藤和智です（その前に既に 92 冊目が出ているのですが、まあ気にしない。あとそろそろ 81 冊目をそろそろ出せ）。

さて、2016 年の「コミックマーケット 90」で、テキストマイニング専門の解説書『Text Mining Maniax：フリーソフトで始める日本語計量テキスト分析』を出してから、その後もテキストマイニング（計量テキスト分析）の解説書を何冊か出してきました（というか、2018 年の『酪 Fan』と、2023 年に開始したシリーズ「SNS 叢書」を除くと、弊サークルの評論同人誌でテキストマイニングが絡まないものを探す方が難しいかもしれません……）。また、評論の他、東方 Project（上海アリス幻樂団）や駅メモ（「ステーションメモリーズ！」モバイルファクトリー）の同人誌でも、テキストマイニングを使った同人誌を多数出すようになっています。

テキストマイニングというと、仮想通貨（暗号資産）の「マイニング」と混同する人もよくいますが、テキストマイニングというのは、文章を単語などの形態素に分けて、そこで得られたデータから、様々な分析をしたり、近年は機械学習などをしたりというものです。そのため、フリーのテキストマイニングソフトである「KH Coder」の製作者である樋口耕一氏などは「計量テキスト分析」という物言いをしているのですが、まあ平たく言えば、文章を対象としたデータ分析や様々なことと考えてもらえれば結構です。

さて、テキストマイニングの同人誌を出していると、この分野に若干詳しい人たちからは、word2vec（ワード・トゥー・ベック）は取り扱っていないのか、という質問をよく受けます。詳しくは本編で解説しますが、word2vec とは、2013 年に提唱された、単語の「意味」を数値化する手法です。そして弊サークルの同人誌では扱ってきませんでした。意図的に避けてきたというよりも、筆者である私が、そういった手法の存在自体は認知していたのですが、ただ自分で試したことがなかったということにつきます。

そもそも弊サークルのテキストマイニングの同人誌は、主として先ほど名前を挙げた KH Coder というツールを使っております。KH Coder では、形態素解析で得られたデータから単語の集計や、クラスター分析や多次元尺度構成法、自己組織化マップといった統計解析など、統計解析の分野では多数の機能をそろえているのですが、機械学習については、ベイズ学習による分類くらいしかなく、そのため疎遠になってしまいました。ちなみに、本編では KH Coder は「出てきません」。

はじめに

弊サークルでは、KH Coder を使う前は（2014 年頃）、RMeCab を使っていました。RMeCab というのは、フリーの統計解析ソフト「R」の解説書で定評がある石田基広氏が製作した、日本語形態素解析エンジン「MeCab」を R で使えるようにするパッケージで、現在でも開発が進められています。2013 年に出した『統計学で解き明かす成人の日社説の変遷：平成日本若者論史 5』（杜の奇跡 20）と『都条例メディア規制の形成：平成日本若者論史 8』（コミックマーケット 84）では、R と RMeCab を使って、N-gram などの KH Coder ではできないような分析をガシガシやっていたのですが、KH Coder を導入してからは、KH Coder でできる分析を中心にいろいろとやるようになったのですが（KH Coder でできない分析を再び導入するようになったのは、2020 年に初めて数回でフェードアウトしてしまったウェブ連載「新・間違いだらけの論客選び REMASTER」で因子分析を用いたときくらいでしょうか）、KH Coder でできない分析からは完全に疎遠になってしまいました。つくづく KH Coder は罪なツールです（褒めていますよ、念のため。KH Coder はテキストマイニングの面白さを理解するためには絶対に、絶対におすすめのツールです！）。

本書では R と RMeCab を用いて word2vec について説明します。ただ、この分野に詳しい方からは、word2vec は機械学習の手法なのだから、R よりも Python を使った方がいいだろう、という意見もあると思います。私も Python による word2vec の作り方は解説書を読みましたが、今回は word2vec という手法に慣れ親しむことと、また弊サークルにおいて R 及び RMeCab によるテキストマイニングの概説書がないということです。一応、『R Maniax』と『Text Mining Maniax』には RMeCab を用いた分析のページはあるのですが、扱いが若干「チョイ役」なので、改めて概説しようと思いました。

私自身 word2vec という手法には不慣れなこともあり、私自身がその手法に親しむという目的もあります。ただ、word2vec という手法は、2023 年夏現在では既に「時代遅れ」のものであり、現在は BERT などのさらに優れた手法も提起されています（2020 年頃に書かれたブログ記事とかだと「定番の手法」などと書かれていたりしますが）。そのことも忘れてはならないと思いますが、他方で ChatGPT のような生成 AI が行政にまで（これ自体はちょっと問題があると思いますが……）浸透している状況において、前時代的でもいいので、手法の一つを学んでおくという意義はあると思います。

word2vec の誕生（提唱）から 10 年であり、今更かよという意見もあるかもしれませんが、一足飛びに最新のツールに飛びついたり、そこから社会のあり方などまで夢想したりというのは技術に幻想を持つ、左右、文理問わない「論客」の悪い癖です。 3

少しでも基礎的、あるいはベースとなるツールや技術を学ぶことこそ、そういった幻想から抜け出すための基本的な態度であり、行為であると思います。

4

目　　次

5

第 1 章 word2vec とは何か

1.1 そもそも word2vec とは？

というわけで本書は、2013 年に提唱された、word2vec という手法について、R や RMeCab を使いつつ説明するというものです。

そもそも word2vec とは、トマス・ミコロフ（Tomas Mikolov）らによって提唱された手法で、単語の意味を「文脈に依存しない」ベクトルの形で表現するものです。ベクトルという言葉が出てきた時点で既に頭が痛くなってしまう人もいるかもしれませんが、まあそこはご理解とご協力を強制いたしますので、我慢してください。

word2vec を説明する際によく使われる例えが、「「king」という単語から「man」を引いて「woman」を足すと「queen」近くなる」というものです（黒橋禎夫『[三訂版] 自然言語処理』（放送大学教育振興会、2023 年）p.83）。ここでは、次のようなモデルを考えます。v() をその単語の持つベクトルだとして、

$$v(\text{仙台市}) - v(\text{宮城県}) + v(\text{岩手県}) \fallingdotseq v(\text{盛岡市})$$

この v(岩手県) を右に移項すると次のようになります。

$$v(\text{仙台市}) - v(\text{宮城県}) \fallingdotseq v(\text{盛岡市}) - v(\text{岩手県})$$

この「仙台市から宮城県を引いたもの」と「盛岡市から岩手県を引いたもの」は、意味から考えれば最も妥当なのが「県庁所在地」でしょう。そのためここでは仙台市のほうだけに注目して

$$v(\text{仙台市}) - v(\text{宮城県}) \fallingdotseq v(\text{県庁所在地})$$

と置き換えて、v(宮城県) を右に移項して左右を入れ替えると、

$$v(\text{宮城県}) + v(\text{県庁所在地}) \fallingdotseq v(\text{仙台市})$$

になります。このように、word2vec には、数値の加減算と意味の加減算が（概ね）一致するという性質があり、これを「加法構成性」といいます（近江崇宏ほか『BERT

による自然言語処理入門』オーム社、2021 年）。

1.2 word2vec の背景：Word Embedding（単語埋め込み）とは何か？

word2vec が生まれた背景にあるものは、Word Embedding（単語埋め込み）という手法です。単語埋め込みとは、単語に対して文脈に依存しない分散表現を与えることで、分散表現とは、文書や単語をベクトルとして表現することを指します。この作業の過程は、ニューラルネットワークを使って何らかの文章（コーパス）から単語の意味を学習するというものになります。

『Deep Learning 2 自然言語処理編』（斎藤康毅：著、オライリージャパン、2018 年）という、Python で word2vec を作るなどのテキストに対する機械学習を解説した書籍では、「You say goodbye and I say hello.」という短いコーパスから word2vec を作るという作業を紹介しています。これに基づいて word2vec に至る道を解説してみます。まず最初の段階として、共起行列というものを作ります。共起行列というのは、例えば先の文章における「you」ならその隣にあるのは「say」、また「say」の隣にあるものは「you」と「goodbye」……というものを、行列として示すものです。ただし、say は 2 回出てくることに注意が必要です。先の短い文章から共起行列を作ると次のようになります（なお対象は文末のピリオドも含めます。）。

you	0,	1,	0,	0,	0,	0,	0
say	1,	0,	1,	0,	1,	1,	0
goodbye	0,	1,	0,	1,	0,	0,	0
and	0,	0,	1,	0,	1,	0,	0
i	0,	1,	0,	1,	0,	0,	0
hello	0,	1,	0,	0,	0,	0,	1
.	0,	0,	0,	0,	0,	1,	0

※斎藤康毅『Deep Learning 2 自然言語処理編』（オライリージャパン、2018 年）p.71

このようにしてできた共起行列から、単語の間の相互情報量（PMI）と呼ばれる値を求めます。求め方については省略しますが、PMI を求める式の中には底が 2 の対数が使われているため、全く共起しない単語 2 つの場合は対数の中の値が 0 になるため、マイナス無限大に発散してしまいます。そのため全く共起しない 2 つの単語につ

いては 0 とする正の相互情報量（PPMI）という値が使われます。先の共起行列なら次のようになるようです。

you	0,	1.807,	0,	0,	0,	0,	0
say	1.807,	0,	0.807,	0,	0.807,	0.807,	0
goodbye	0,	0.807,	0,	1.807,	0,	0,	0
and	0,	0,	1.807,	0,	1.807,	0,	0
i	0,	0.807,	0,	1.807,	0,	0,	0
hello	0,	0.807,	0,	0,	0,	0,	2.807
.	0,	0,	0,	0,	0,	2.807,	0

※斎藤、前掲 p.81

ただ、この行列は 0 になる要素が非常に多いので、行列の横の要素をまとめます。これを次元削減といいます。同書では、最終的に 2 次元までカットしています。ここまでがカウントベースの手法ですが、このような手法はコーパスが膨大になったときには時間がかかります(全ての単語について計算を行うため)。そこで、ニューラルネットワークを使って推論ベースの手法を行います。推論ベースの手法というのは、例えば先の「You say goodbye and I say hello.」という文章の場合、「You ○ goodbye and I say hello.」と「say」を空白にして、前後の単語から、どういう単語が来るのかということを推測する手法です。手法の流れは、

1. それぞれの単語に ID を付けて、one-hot 表現と呼ばれる表現に変換する。例えば、you の単語 ID を 0 とすると、このコーパスには全部で 7 個の単語があるので、you は (1,0,0,0,0,0,0)、goodbye は (0,0,1,0,0,0,0) などというようになる。
2. この 2 つのベクトルを入力層（コンテキスト）として、中間層を経由して出力層を算出する。ニューラルネットワークは入力層 (2 つ) →中間層→出力層 (1 つ) という流れを行列計算を用いて行う。そしてこの中間層が分散表現である。
3. この場合、出力層では say を指す (0,1,0,0,0,0,0) が出力されればいいので（ターゲット)、後は全てのターゲットに対応するコンテクストを使用して、ターゲットが出る確率が高くなるように学習させる。

※詳しくは、斎藤、前掲第 3 章を参照。

8　ちょっと説明が冗長になってしまっているので、詳しくは同書を読んで、可能なら

ば Python でコードを組んでほしいのですが、いずれにせよコーパスからコンテキストとターゲットを設定して、ターゲットの出る確率が高くなるようにニューラルネットワークを使って学習させる、という方法は、先に見たようなカウントベースの手法よりも早く計算できるようになります。このモデルを CBOW モデルといいます。

1.3 word2vec 作成のモデル

前節では CBOW モデルを紹介しましたが、逆に Skip-Gram モデルという、CBOW モデルを逆にしたようなモデルも存在します。これは、先の斎藤本における「You say goodbye and I say hello.」というコーパスをベースに考えると、「○ say ○ and I say hello.」というものを仮定して、say という単語の前後に何が出てくるのか、ということを推測するものです。この場合はターゲットは you と goodbye の 2 つになります。

1.4 そもそも自然言語処理とは？

word2vec の作成などの文章に対する学習や解析などは、自然言語処理という研究分野の一つです。自然言語処理という言葉だけなら多くの人が聞いたことがあるかもしれませんが、そもそも「自然言語」というのは、プログラミング言語などの人工言語（ある目的のために人工的に作られた言語）とは違い、人類史の中で自然発生的に生まれた言語のことを指します（黒橋、前掲 p.9）。日本語や英語、中国語、韓国語、ドイツ語、フランス語、アラビア語などがありますが、単に「言語」といったときには自然言語のことを指します。

言語というものはコミュニケーション、思考、記録の道具としての側面があり、また曖昧性を持つものです。そのため自然言語をコンピュータで扱うときには、特に後者が難しいものとして立ちはだかりました。そのため膨大なコーパスなどで対応する必要がありましたが、これがニューラルネットワークの導入により大きく進んだとされています（黒橋、前掲 pp.10-12）。

自然言語処理にはいくつかの段階があります。その中でも最も基本的なものが形態素解析です。これは、弊サークルの同人誌でも度々説明しているとおり、文章を単語などの形態素に分けて単語を認識するものです。形態素解析もまた、自然言語処理の一つです。

しかし自然言語処理はこれでは終わりではありません。日本語の場合、「構文解析」「格解析」「照応・省略解析」「談話構造解析」、合計 5 つが必要になるとされています（黒橋、前掲 p.13）。「構文解析」とは、いわゆる係り受けの解析で、文中の語句の修飾関

係を解析するものとされています。係り受け解析のツールとしては、フリーの係り受け解析ソフトである「Cabocha」（南瓜。https://taku910.github.io/cabocha/）があります。また、「照応・省略解析」「談話構造解析」は、文をまたがる語句や節・文の結びつきを解析するものとされており（黒橋、前掲 p.13）、コンピュータにやらせるにはかなり高度なものです。機械学習の分野ではありませんが、近年は三木那由他『会話を哲学する：コミュニケーションとマニピュレーション』（光文社新書、2022 年）や中井陽子ほか『会話データ分析の実際』（ナカニシヤ出版、2022 年）などの解説書があるので、周辺の知識として読んでみても損はないでしょう。

自然言語処理には、形態素解析のほか、文章の自然さを確率によって表現する「数理モデル」、文章から固有名詞や日付や数値表現を抽出する「固有表現抽出」、文章の意味の類似度を定量的に評価する「文章の類似度比較」などが挙げられます、さらに発展的なものとして、文章の分類や、ChatGPT のような文章の生成、そして文章の校正があります（近江ほか、前掲 pp.2-3）。

1.5 word2vec の問題点と、より発展的なモデル

word2vec にはいくつかの問題点が指摘されています。まず、word2vec は文脈に依存しない形で単語に対して一意に意味（ベクトル）を与えるため、「彼は舞台の上手（かみて）に立った」と「彼は料理上手（上手）だ」という 2 つの文章における「上手」が区別できないということと、「甲が乙に本を貸した」「乙が甲に本を貸した」といったものに同一の分散表現が与えられるように、語順を考慮しないことが挙げられています（近江ほか、前掲 p.20）。

2018 年に Google によって発表されたモデルである BERT は、Attention という手法を使うことによって、離れた位置にある情報も取り入れられるようになり、より深い文脈の考慮ができるようになったとされています（近江ほか、前掲 p.5）。

word2vec にしろ BERT にしろ、機械学習による自然言語処理のためには、事前に学習させたデータが必要になります。自分でも作ることはできますが、世の中には様々な事前学習データがあるので探してみてもいいでしょう（長崎俊紀『Word2Vec 自然言語処理活用ハンドブック』（インプレス、2022 年）の第 2 章の章末に主要なものが紹介されています）。また word2vec や BERT などの自然言語処理ツールの歴史については、本章でも引用・参照文献として使っている、黒橋禎夫『[三訂版] 自然言語処理』（放送大学教育振興会、2023 年）や近江崇宏ほか『BERT による自然言語処理入門：Transformers を使った実践プログラミング』（オーム社、2021 年）がわかりやすいです。

10

第 2 章 RMeCab の導入と操作

2.1 R の導入

本書では、R と RMeCab を使って word2vec を作る作業を紹介します。そのためにはまず R と RMeCab を導入する必要があります。R の導入については他の同人誌でも解説しているのですが、改めて紹介します。

R は、オープンソースで開発されているフリーの統計解析言語で、2000 年代からデータサイエンスなどでアカデミックな分野でも多く使用されてきましたが、近年では機械学習分野で勝る Python が地位を高めており、逆に R の地位は低くなっています。それでも、統計解析やデータサイエンス分野では Python よりも優れている面は存在し、機械学習をそれほど伴わない分野であればいまだに現役だと思います。

R は本拠地である「R-project」のサーバー「CRAN」か、または世界中で（もちろん日本にも）作られている R のミラーサイトからインストールする必要があります。Google などで「R 言語」とか「R project」などと検索して、R の本拠地のサイト（https://www.r-project.org/ ）に行き、左側の「CRAN」をクリックします。そうすると世界各地の CRAN へのリンクがあるので、お好きなところを選んで、移動した先で「Download R for Windows」をクリックし、さらに「base」、さらに「Download R-〇.〇.〇 for Windows」をクリックして、最新の R をダウンロードします。

なお、R に限らず、MeCab や LaTeX、Python などもそうですが、この手のオープンソースのソフトウェアをインストールするときには、Windows フォルダの中の「Program Files」ではなく、C ドライブ直下に「usr」、さらにその中に「local」というフォルダを作って、そこにインストールするという方法をおすすめしております。特に MeCab は、本書では扱いませんが、ユーザー辞書による MeCab のカスタマイズや、NEologd、unidic などの導入などは、MeCab が Program Files フォルダの中にあるととてもやりづらいので、Program Files の外でやった方が効率がいいのです。

R はインストーラーの通りにインストールすれば問題はありません。ただ、R は様々な統計解析で使えるのですが、高度な統計解析のためにはパッケージが必要になりますので、R をインストールしたら、パッケージも一通りインストールしておきましょう。

まず R を起動して、「パッケージ」から「ダウンロードサイトの選択」を開き、Ctrl キーを押しながらクリックすることで少なくとも「CRAN」「CRAN(extra)」「R-forge」

11