

Word2vec を用いた文章構造の解析手法

プロジェクトマネジメントコース

ソフトウェア開発管理グループ

矢吹研究室

1442069

須山武弘

目次

第 1 章	序論	2
第 2 章	背景	3
2.1	本研究の全体的背景	3
2.2	自然言語	4
2.3	自然言語処理	5
2.4	Word2vec について	6
第 3 章	目的	7
第 4 章	手法	8
4.1	本研究全体の流れ	8
4.2	Linux について	9
4.3	使用・インストールするツール	10
4.4	環境構築	30
4.5	Word2vec	40
第 5 章	結果	53
第 6 章	考察	57
第 7 章	結論	61
	謝辞	62

第 1 章

序論

レポートや論文を書く際には、読みやすく、論理的な文章を書くことが大切である。論理的な文章を書くための書き方として、世界で標準的なパラグラフ・ライティング (Paragraph writing) がある。パラグラフ・ライティングは、英語文章の一般的スタイルであり、序論、本論、結論の 3 部構成となっている。序論でトピックとなる文が示され、本論は序論に続く支持文となり、最後に結論で文章をまとめる。冒頭にトピックとなる文章を示すと伝えたいことが明確になり、速読が可能となったり、内容の理解が深まるなど多数のメリットがある。

言語を定量的に表すツールとして、Word2vec がある。Word2vec は、単語をベクトルへ変換することができるため、文章の話題の方向性を解析し、文章作成の補助ができるのではないかと仮説を立て、本研究に取り組んだ。

第 2 章

背景

2.1 本研究の全体的背景

近年，コンピュータは人間と対話するようになってきている．ソフトバンク社の開発した，感情認識ヒューマノイドロボットの Pepper を始めとするロボットや，Amazon 社の発売する AI スピーカー Echo，身近な存在ではスマートフォンの音声認識システムもそうである．また，これから大きく普及するであろう IoT（Internet of Things：モノをインターネットに繋ぐ技術）技術を用いた様々なものにも音声認識システムは搭載されるであろう．このことから，コンピュータは人間が自然に発する言語である自然言語を理解し，処理，出力することが要求されている．このような，コンピュータにおける自然言語処理の必要性が高まっていることに注目し，コンピュータと自然言語でなにか研究はできないかと考えた．Word2vec を用いてコンピュータに処理をさせ，自然言語を数値化し，文書構造を解析させることを考えた．

2.2 自然言語

自然言語とは、人間が日常の意思疎通のために用いられ、自然に発せられる言語のことである。人間が自然に発する言語ということもで、人それぞれの出身やできごとなどの文化的背景などが絡み、曖昧な表現が含まれる。そのため、プログラム言語などの数学的言語とは違い、曖昧さを含んでいる自然言語は直接コンピュータが認識することはできない。このことから、コンピュータへ自然言語を認識させるには自然言語処理が必要である。

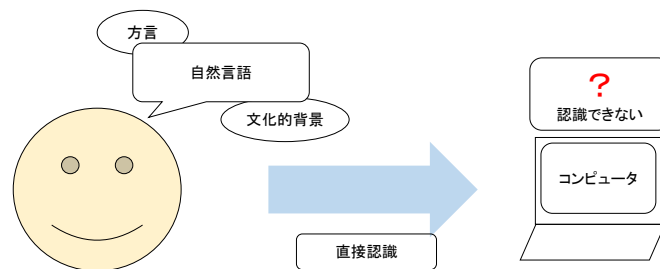


図 2.1 自然言語は直接認識できない

2.3 自然言語処理

自然言語処理とは、前述した自然言語をコンピュータへ認識させるために行う処理のことである。

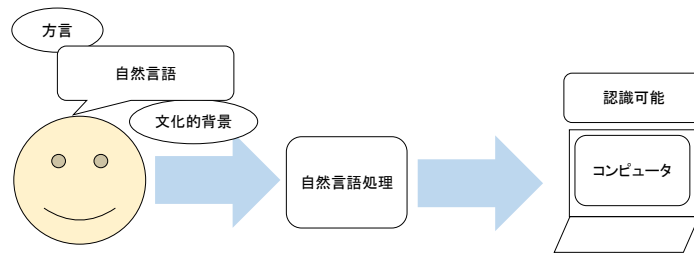


図 2.2 コンピュータへの認識

2.4 Word2vec について

本研究で用いる Word2vec は、テキスト処理を行うニューラルネットワークのことである。コーパスを入力すると、単語の特徴量ベクトルが出力される。つまり、テキストを数値化することができる。

また、Word2vec は類似後のベクトルをベクトル空間にグループ化することができるので、数値に基づいて類似性を判断することができる。

2.4.1 コーパスとは

コーパスとは、自然言語処理の研究に用いるため、自然言語の文章を構造化し、大規模に集積したものである。Word2vec では、MeCab を利用し、形態素解析を行ったあと、コーパスを作成することができるが、Wikipedia など、大量のデータを利用してコーパスを作成するには非常に時間がかかる。

第 3 章

目的

Word2vec を用いて文字列である文章をベクトルへ変換し，定量的に文章構造を解析することでパラグラフ・ライティングができているかを調査する．

第 4 章

手法

4.1 本研究全体の流れ

以下の手順で研究を進めた．

1. 仮想環境の導入．
2. Word2vec 環境構築．
3. 形態素解析（MeCabwo 使用）
4. 日本語 Wikipedia エンティティベクトルのコーパスを使用し，Word2vec によって文章をベクトルへ変換．
5. R の導入．
6. 多数の文章で主成分分析．

4.2 Linux について

Linux とは，Windows や MacOS と同じ OS(オペレーティング・システム) のことである．Linux は，Unix が源流のオープンソース OS である．オープンソースなので，無料で使え，改変可能なプログラムである．

Linux を使うにあたって，様々な特徴がある．

1. 無料である．
2. オープンソースのため，改変が可能である．
3. プログラムを動かす環境として優れている．
4. 低スペックなコンピュータでも動作可能である

4.3 使用・インストールするツール

本研究では仮想環境を用いて研究を行ったため、仮想環境の構築手法を記す。

4.3.1 chocolatey

今回、環境構築に必要なものをインストールするにあたり、chocolatey をインストールする。これは、必要なソフトがまとめてインストールが可能であるため、これを利用しソフトウェアをインストールする。

chocolatey の大まかなインストール手順は以下の通りである。

1. chocolatey の Web ページへ行き、インストールコマンドをコピーする。
2. 管理者権限のあるコマンドプロンプト（PowerShell）を起動し、ペーストする。
3. インストールするもののコマンドを入力する。

「chocolatey」と web 検索をするか、

<https://chocolatey.org/>

の URL へアクセスすれば、chocolatey の公式ページへ飛ぶことができる。

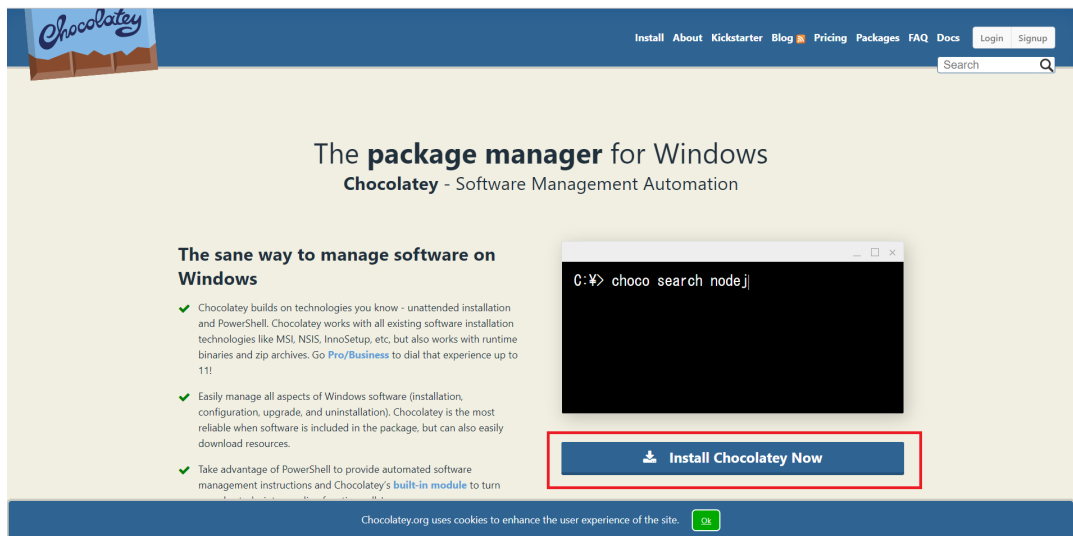


図 4.1 chocolateyHP

上図の赤枠部へアクセスすると、インストール画面へ遷移する。

インストールページへ遷移したら、少し下へスクロールをする。そうすると、赤枠部のようなテキストが見えてくる。

コマンドプロンプトを利用する場合は赤枠部上、PowerShell を利用している場合は下のテキストをコピーし、管理者権限のあるコマンドプロンプトにペーストする。

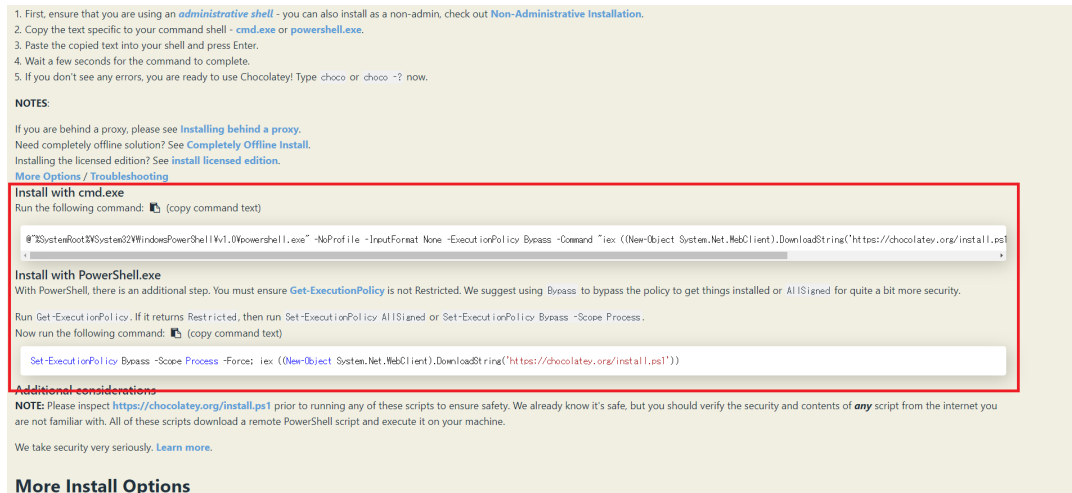
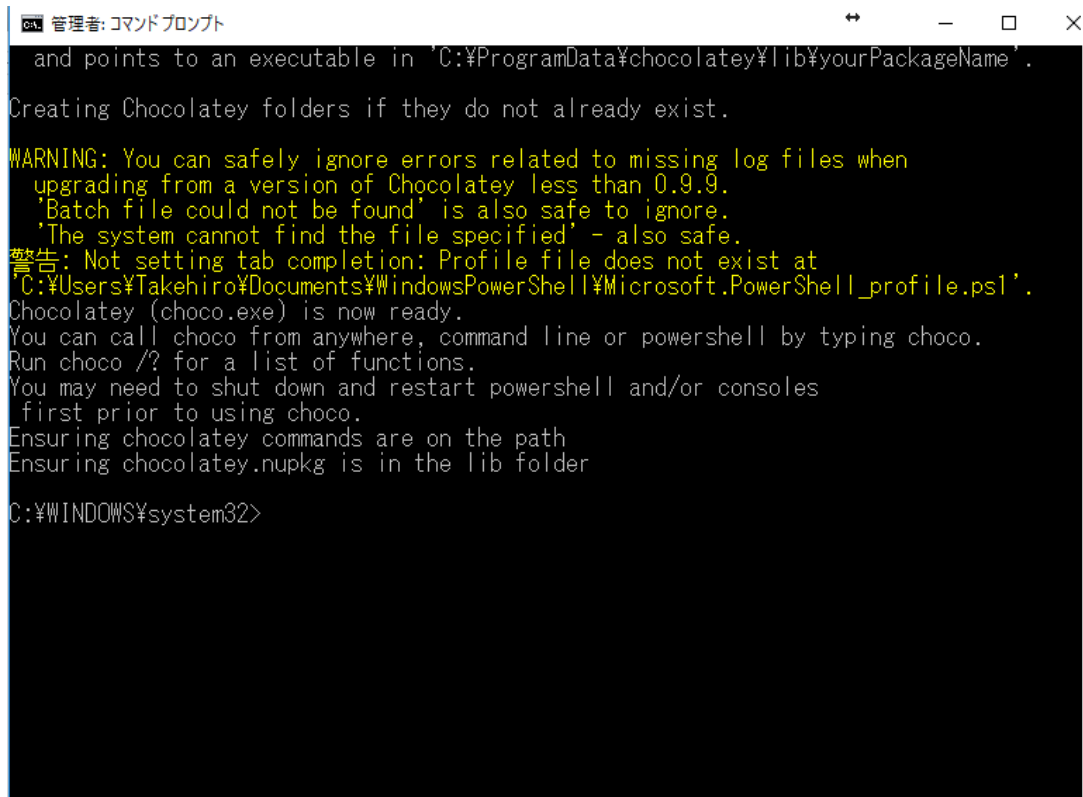


図 4.2 chocolatey インストールページ

管理者権限のあるコマンドプロンプト (PowerShell) を起動する。ここで、管理者権限のないコマンドプロンプトを開いてしまわぬよう注意すること。

先程のコピーしてきたコマンドをペーストし、以下のようにになれば成功である。



```
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.
Creating Chocolatey folders if they do not already exist.
WARNING: You can safely ignore errors related to missing log files when
         upgrading from a version of Chocolatey less than 0.9.9.
         'Batch file could not be found' is also safe to ignore.
         'The system cannot find the file specified' - also safe.
警告: Not setting tab completion: Profile file does not exist at
      'C:\Users\Takehiro\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder

C:\WINDOWS\system32>
```

図 4.3 chocolatey インストール

他にインストールをしたいものがある場合は、以下のページにアクセスするとインストールできるパッケージの一覧が見れる。インストールしたいパッケージのコマンドを管理者用コマンドプロンプトへ入力するとインストールできる。

<https://chocolatey.org/packages>

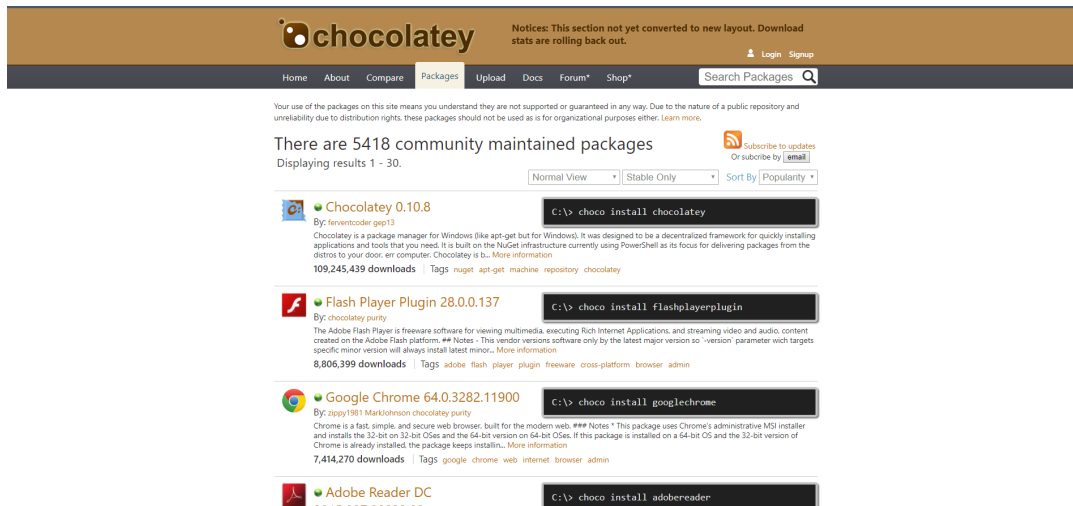


図 4.4 chocolate パッケージ

4.3.2 VirtualBox

本研究では，Windows 上で Linux 環境を構築するために VirtualBox を使用する．本来，LinuxOS を扱いたいが，WindowsOS のマシンの為，VirtualBox を使用し，使用しているマシン上に仮想的なマシンを作成し，別の OS をインストール・実行できるようにする．

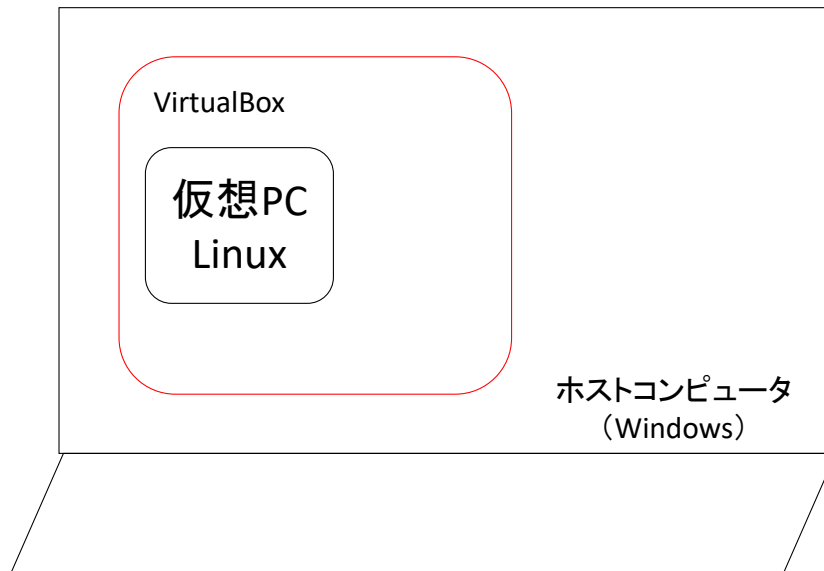


図 4.5 VirtualBox 概要

VirtualBox はコンピュータ上で直接動作している通常の OS にとってはアプリケーションの一つであり，ほかのソフトと同じように起動することができるものである．起動すると仮想的なマシンが構築され，元の OS とは独立に別の OS を起動することができる．VirtualBox が実行されている OS をホスト OS，VirtualBox 上で実行されている OS をゲスト OS という．

元は独立系のソフトウェア企業が開発・販売していた製品だったが，開発元が Sun Microsystems 社に買収され，その後同社が Oracle 社に買収されたため，Oracle 社が開発元となり，正式名称も「Oracle VM VirtualBox」となっている．さらに，VirtualBox 本体は GPL に基づいたオープンソースソフトウェアとして公開され，だれでも自由に入手・利用・改変・再配布などが行える．

VirtualBox を使う上での注意点として，現時点での VirtualBox は仮想メモリをサポートしていないため，実メモリ以上のメモリを仮想 PC が使用することはできない．仮想メモリを使うと動作が遅くなるため，仮想 PC には実メモリ以内のサイズを割り当てる．そのため，仮想 PC を 1 台だけ起動するのであれば問題ないが，複数の仮想 PC を同時に起動させる場合これがネックになってしまう．同時起動させる全ての仮想 PC のメモリサイズの合計が実メモリのサイズを超えないようにする．そのため，VirtualBox をインストールする PC には多くのメモリが必要で，最低 4GB 以上の PC を使うようにすべきである．本研究に用いる際もかそう PC のメモリは 2 GB では足りないので，注意が必要である．

次に，chocolatey を使用した VirtualBox のインストール方法を説明する．

下図のように

```
choco install virtualbox
```

とコマンドを入力すると，インストールすることができる．

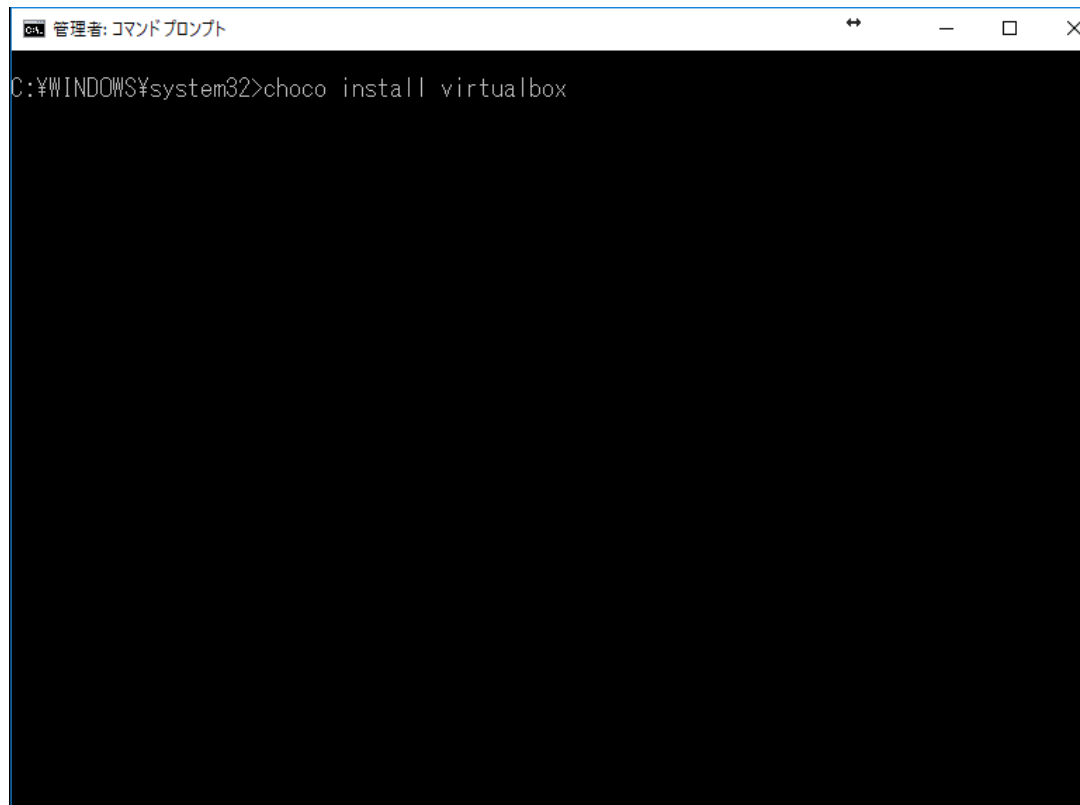


図 4.6 chocolatey VirtualBox のインストール

実際に VirtualBox を起動し，図 4.7 のようになれば成功である．

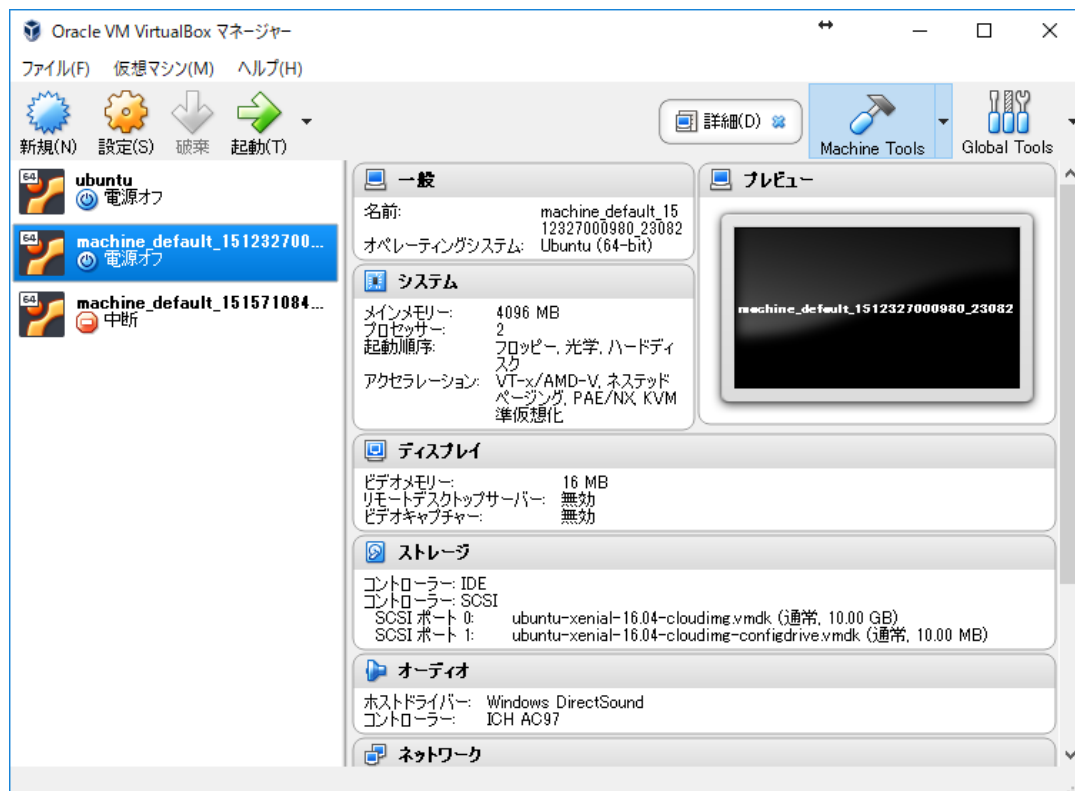


図 4.7 chocolatey VirtualBox のインストール

VirtualBox の用語

ホストマシン

物理的に存在するコンピュータのこと。

ホスト OS

ホストマシンにインストールされている OS のこと。VirtualBox はホスト OS にインストールされる。

バーチャルマシン (仮想マシン)

VirtualBox が作成する論理的なマシンのこと。ゲストマシンに割り当てるために、VirtualBox がホストマシンのコンピュータ資源 (CPU やメモリ、HDD 等) の一部を仮想化する。ホストマシンの資源を使い切らない限り、ゲストマシンを複数作成したり、多重起動させることができる。

ゲスト OS(仮想 OS)

ゲストマシンにインストールされる OS のこと。

仮想ディスク

ゲストマシンが使用する仮想のハードディスクのこと。バーチャルマシンからはこれを物理ディスクとして扱うことができる。仮想ディスクの実態はホストマシン内にファイルとして存在する。

4.3.3 Vagrant

Vagrant とは、仮想環境を簡単に構築・管理することができるツールである。Vagrant を使用することにより、仮想マシンの設定や作成などが容易にできる。

Vagrant を用いるにあたって、様々な利点がある。

1. コマンド一つで仮想マシンを作成することができる。
2. ホストマシンの環境に依存せずに開発やテストができる。
3. Vagrantfile に構成を記述できるので、環境構築が楽になる。

Vagrant のインストール

Vagrant は chocolatey のコマンドでインストールすることができる。

コマンドは以下の通りである。管理者権限のあるコマンドプロンプトに

`choco install -y vagrant`

と入力し、実行する。

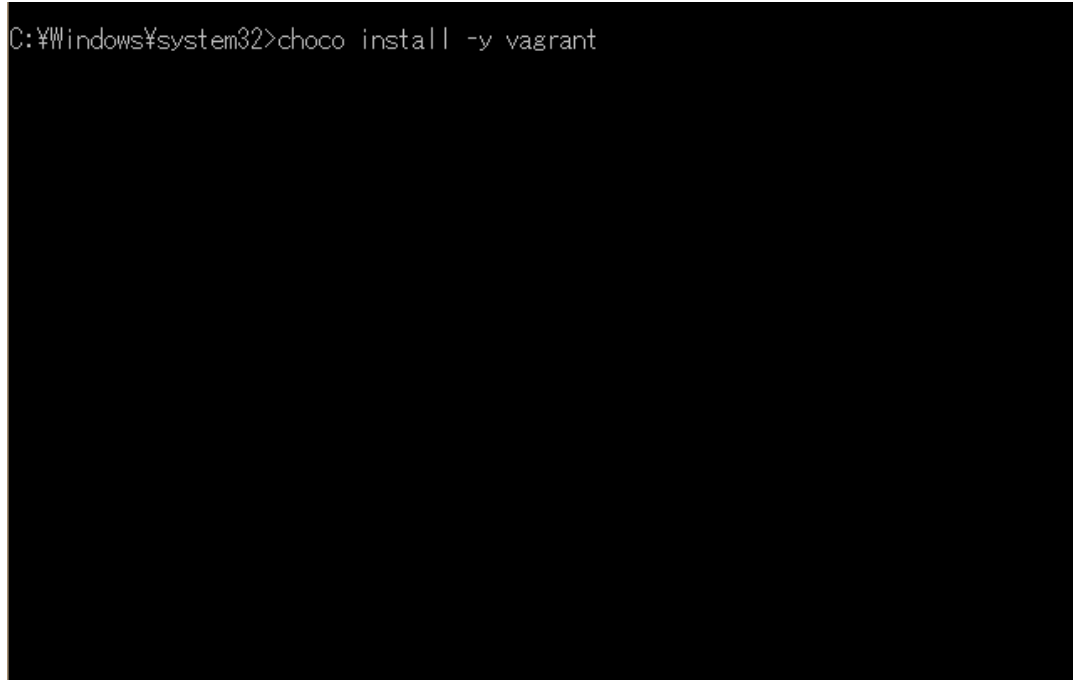


図 4.8 chocolatey Vagrant のインストール

図 4.9 のようになればインストール成功である。

```
C:\Windows\system32>choco install -y vagrant
Chocolatey v0.10.3
Installing the following packages:
vagrant
By installing you accept licenses for the packages.
vagrant v1.8.1.20160318 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages. 0 packages failed.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- vagrant - vagrant v1.8.1.20160318 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

C:\Windows\system32>
```

図 4.9 chocolatey Vagrant のインストール完了

Vagrant 起動の仕方

コマンドプロンプトを起動し, " /Vagrant/machine " までディレクトリを変更する .
やり方は以下の通りである .

" cd /vagrant/machine "

と入力し , 実行する . cd とは , チェンジディレクトリの意味である .

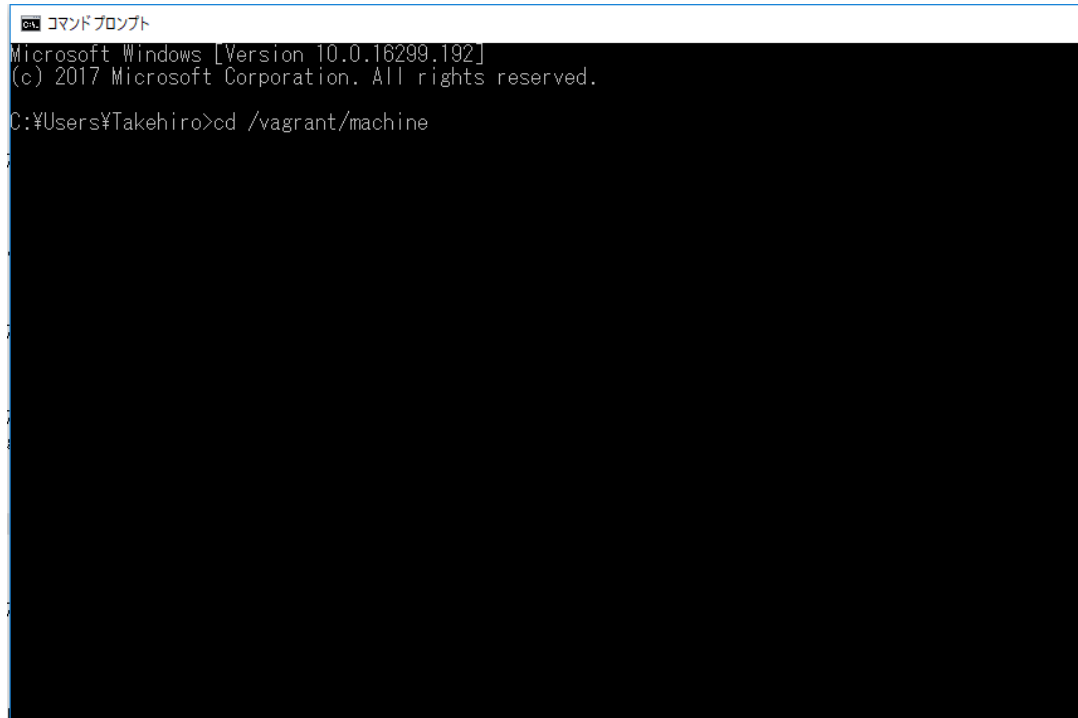
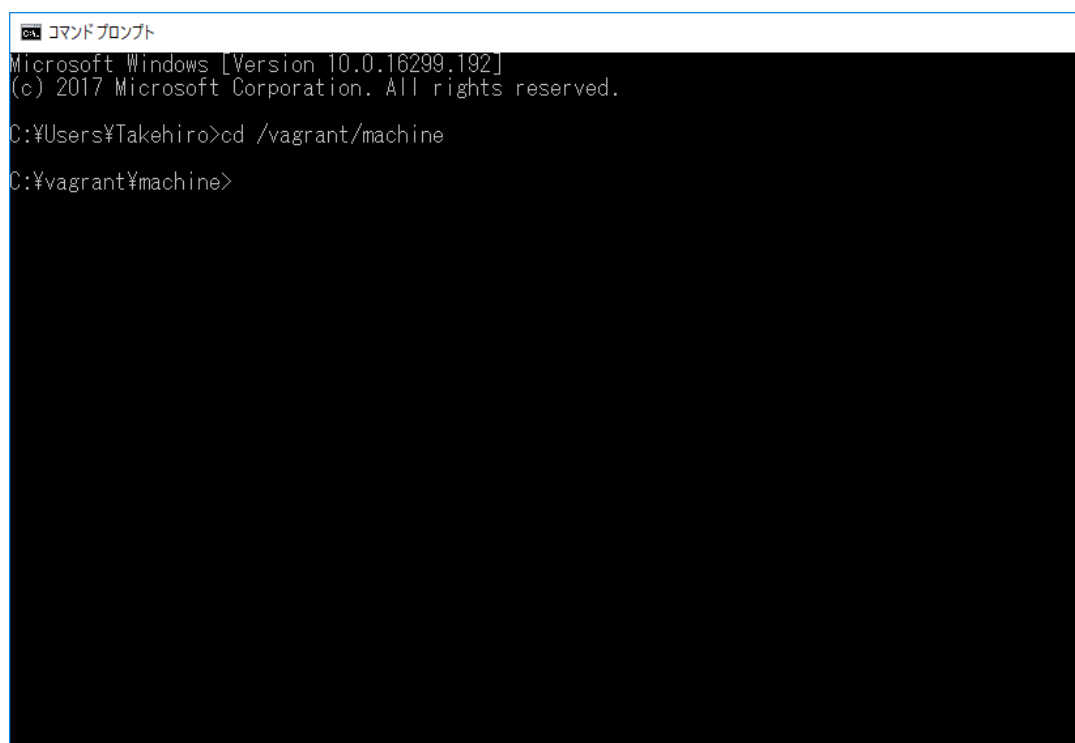


図 4.10 Vagrant 起動 ディレクトリ変更

以下のようなになればディレクトリの変更に成功している .

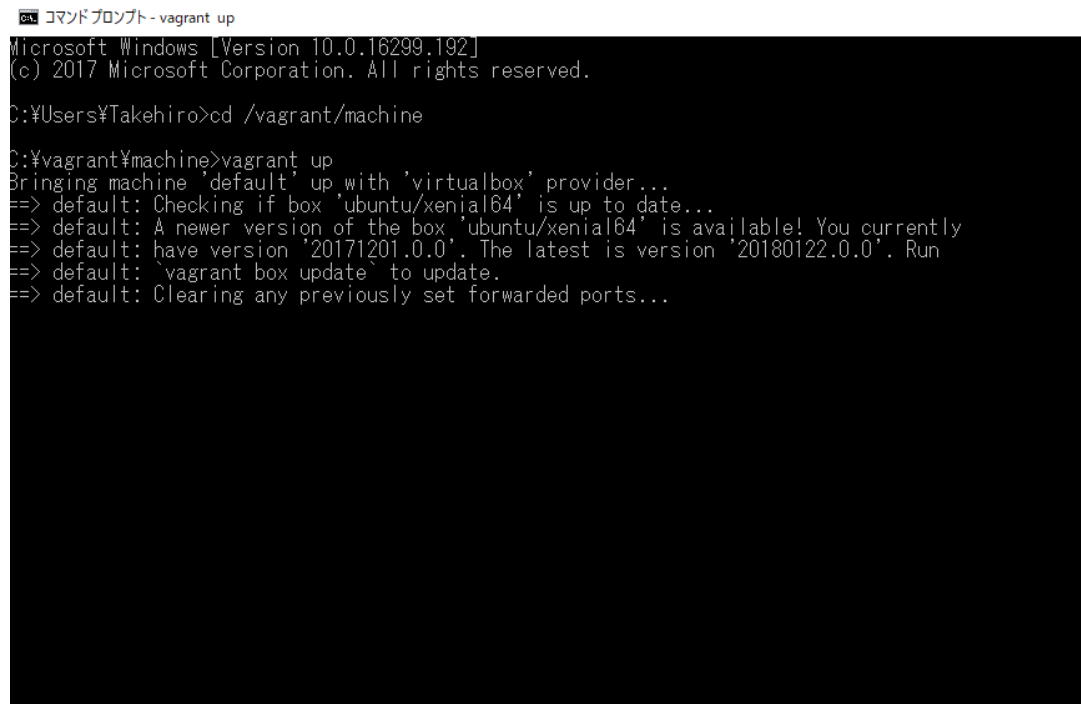


```
コマンドプロンプト
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\¥Takehiro>cd /vagrant/machine
C:\vagrant¥machine>
```

図 4.11 Vagrant 起動 ディレクトリ変更完了

ディレクトリの変更が済んだら、マシンの起動コマンドを入力する。起動コマンドは、”
vagrant up ” である。

A screenshot of a Windows command prompt window titled "コマンドプロンプト - vagrant up". The window shows the following text: "Microsoft Windows [Version 10.0.16299.192] (c) 2017 Microsoft Corporation. All rights reserved. C:\Users\Takehiro>cd /vagrant/machine C:\vagrant\machine>vagrant up Bringing machine 'default' up with 'virtualbox' provider... ==> default: Checking if box 'ubuntu/xenial64' is up to date... ==> default: A newer version of the box 'ubuntu/xenial64' is available! You currently ==> default: have version '20171201.0.0'. The latest is version '20180122.0.0'. Run ==> default: `vagrant box update` to update. ==> default: Clearing any previously set forwarded ports..."

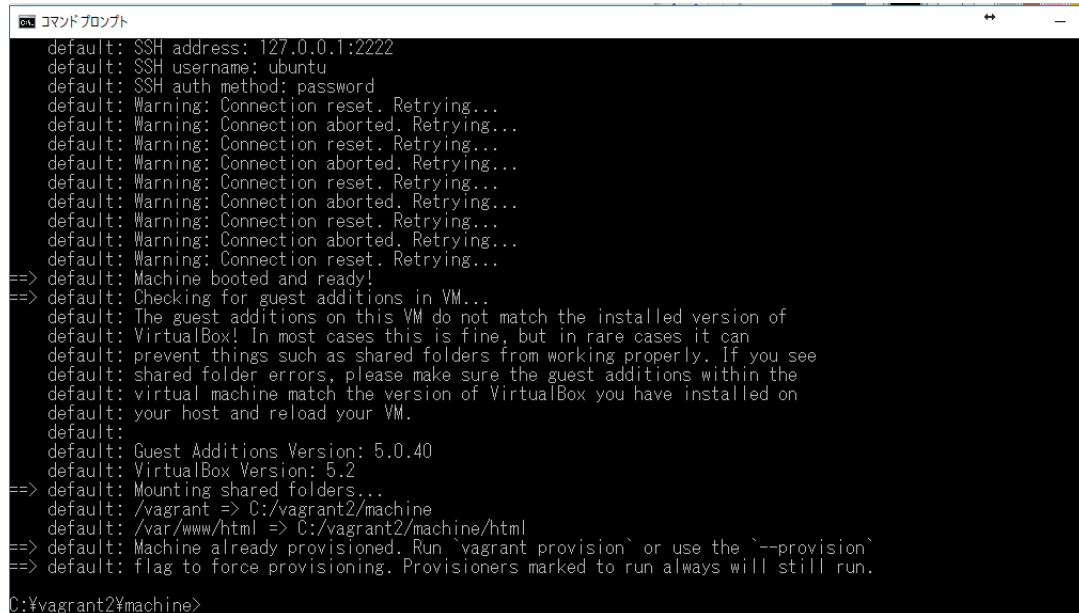
```
コマンドプロンプト - vagrant up
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Takehiro>cd /vagrant/machine

C:\vagrant\machine>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'ubuntu/xenial64' is up to date...
==> default: A newer version of the box 'ubuntu/xenial64' is available! You currently
==> default: have version '20171201.0.0'. The latest is version '20180122.0.0'. Run
==> default: `vagrant box update` to update.
==> default: Clearing any previously set forwarded ports...
```

図 4.12 Vagrant 起動

以下のようになれば起動完了である．なお，初回起動には時間がかかるので暫く待つ必要がある．



```
コマンドプロンプト
default: SSH address: 127.0.0.1:2222
default: SSH username: ubuntu
default: SSH auth method: password
default: Warning: Connection reset. Retrying...
default: Warning: Connection aborted. Retrying...
default: Warning: Connection reset. Retrying...
default: Warning: Connection aborted. Retrying...
default: Warning: Connection reset. Retrying...
default: Warning: Connection aborted. Retrying...
default: Warning: Connection reset. Retrying...
default: Warning: Connection aborted. Retrying...
default: Warning: Connection reset. Retrying...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
default: The guest additions on this VM do not match the installed version of
default: VirtualBox! In most cases this is fine, but in rare cases it can
default: prevent things such as shared folders from working properly. If you see
default: shared folder errors, please make sure the guest additions within the
default: virtual machine match the version of VirtualBox you have installed on
default: your host and reload your VM.
default: Guest Additions Version: 5.0.40
default: VirtualBox Version: 5.2
==> default: Mounting shared folders...
default: /vagrant => C:/vagrant2/machine
default: /var/www/html => C:/vagrant2/machine/html
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
C:\vagrant2\machine>
```

図 4.13 Vagrant 起動完了

仮想マシンへのログイン方法は下記の通りである .

vagrant up を実行したところへ

" vagrant ssh "

と入力する . 以下のようにになれば成功である .

```
C:\vagrant2\machine>vagrant ssh
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-109-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

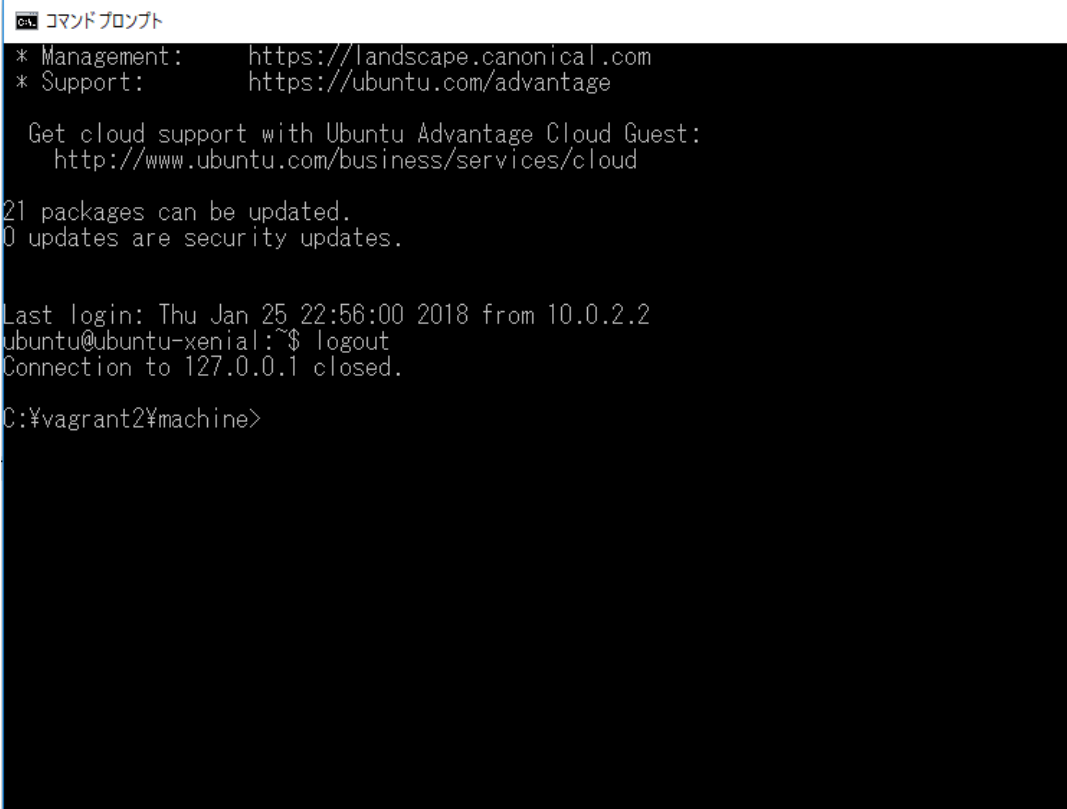
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

21 packages can be updated.
0 updates are security updates.

Last login: Thu Jan 25 22:34:27 2018 from 10.0.2.2
ubuntu@ubuntu-xenial:~$
```

図 4.14 VagrantSSH

ログアウト方法は、「Ctrl + D」でログアウトする。以下のようにになれば成功である。



```

❏ コマンドプロンプト
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

21 packages can be updated.
0 updates are security updates.

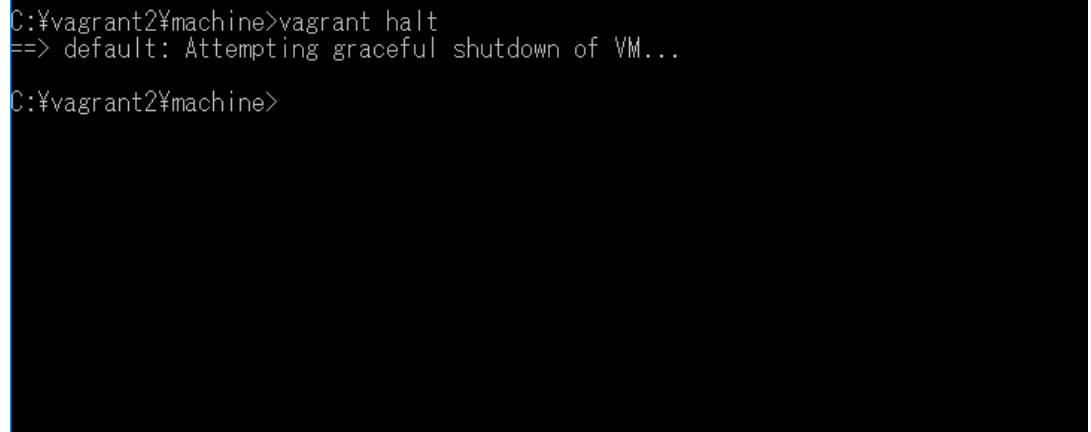
Last login: Thu Jan 25 22:56:00 2018 from 10.0.2.2
ubuntu@ubuntu-xenial:~$ logout
Connection to 127.0.0.1 closed.
C:\vagrant2\machine>
```

図 4.15 Vagrant ログアウト

だが、ログアウトしただけで仮想マシンの電源は切れていない。

そこで、電源を切るコマンドを下記の通り実行し、図 4.16 のようになれば成功である。

” vagrant halt ”



```
C:\vagrant2\machine>vagrant halt
==> default: Attempting graceful shutdown of VM...
C:\vagrant2\machine>
```

図 4.16 Vagrant シャットダウン

Vagrant コマンド

`vagrant up`

仮想マシンの起動を行う。

`vagrant ssh`

仮想マシンの再起動を行う。

`vagrant halt`

仮想マシンの停止を行う。

`vagrant destroy`

仮想マシンの削除を行う。

`vagrant ssh`

仮想マシンにログイン。

4.4 環境構築

本研究では、VirtualBox、Vagrant を使用し、仮想環境を構築した。これまでの説明は、本研究に用いるツールの基本的インストール方法や、基本的知識である。ここからは、本研究で実際に用いた環境設定を書く。なお、chocolatey、VirtualBox、Vagrant がインストール済みであることが前提となっている。

本研究においては、矢吹研究室公式の仮想マシンを使用する。

まず、管理者権限でないコマンドプロンプトを起動し、Guest Addition の更新、ディスクサイズの変更を簡易化するためのプラグインを導入する。

それぞれ，以下のようになれば成功である．

```
C:\>vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Installed the plugin 'vagrant-vbguest (0.15.0)'!

C:\>
```

図 4.17 Guest Addition の更新

```
C:\>vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Installed the plugin 'vagrant-vbguest (0.15.0)'!

C:\>vagrant plugin install vagrant-disksize
Installing the 'vagrant-disksize' plugin. This can take a few minutes...
Installed the plugin 'vagrant-disksize (0.1.2)'!

C:\>
```

図 4.18 ディスクサイズの変更を簡易化するためのプラグインを導入

次に仮想マシンを用意する．先ほどと同じく，管理者でないコマンドプロンプトに以下のコマンドを実行する．コマンドの意味も下記の通りである．

```
cd /
```

ルートディレクトリへ変更．

```
mkdir vagrant
```

vagrant というフォルダを作成．

以下のように何も出てこなければ成功である．念のため，図 4.20 のように作成されているか確認すると良い．

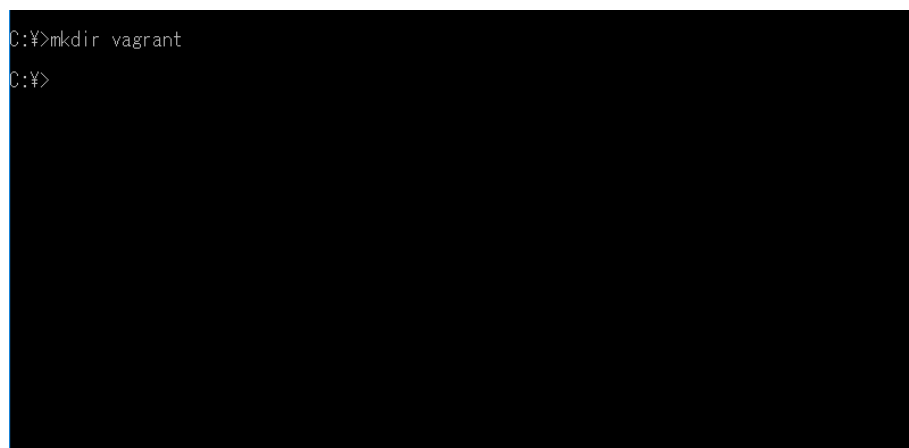


図 4.19 フォルダ作成

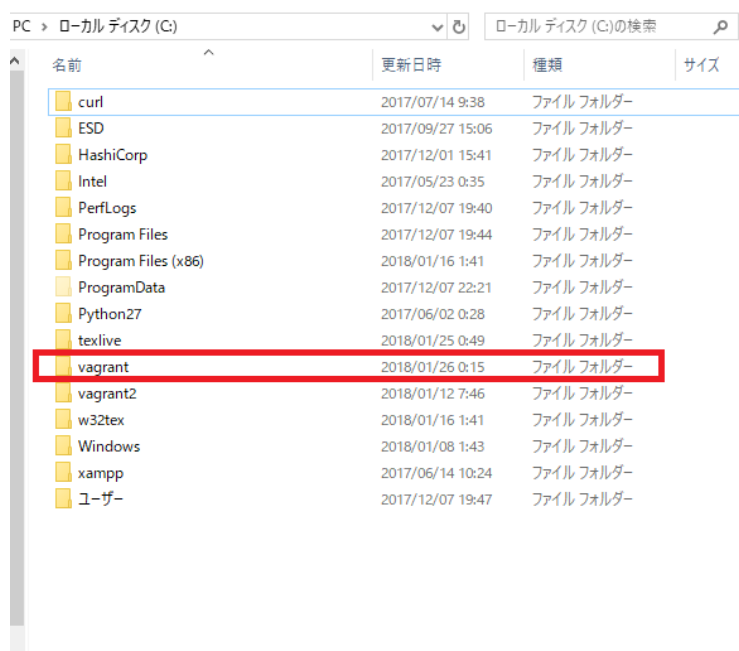


図 4.20 フォルダ作成 確認

```
cd /vagrant
```

先ほど作成した vagrant というフォルダ内へ移動

```
git clone https://github.com/yabukilab/machine.git
```

矢吹研究室公式マシンの github とクローンする .

以下のようなになればクローン成功である .

```
C:\¥>cd /vagrant  
  
C:\¥vagrant>git clone https://github.com/yabukilab/machine.git  
Cloning into 'machine'...  
remote: Counting objects: 127, done.  
remote: Compressing objects: 100% (6/6), done.  
Remote: Total 127 (delta 2), reused 5 (delta 2), pack-reused 119  
47/127)  
Receiving objects: 100% (127/127), 23.32 KiB | 1.37 MiB/s, done.  
Resolving deltas: 100% (56/56), done.  
C:\¥vagrant>
```

図 4.21 矢吹件公式マシン クローン

次に、下記コマンドで machine ディレクトリへ移動する。

```
cd machine
```

本研究では、メモリの容量が不足してしまうため、仮想マシンの起動をする前に Vagrantfile の設定を変える必要がある。メモリ容量の変更方法は下記のとおりである。

C:/vagrant/machine の中に Vagrantfile というファイルが存在する。このファイルをテキストエディタで開く。

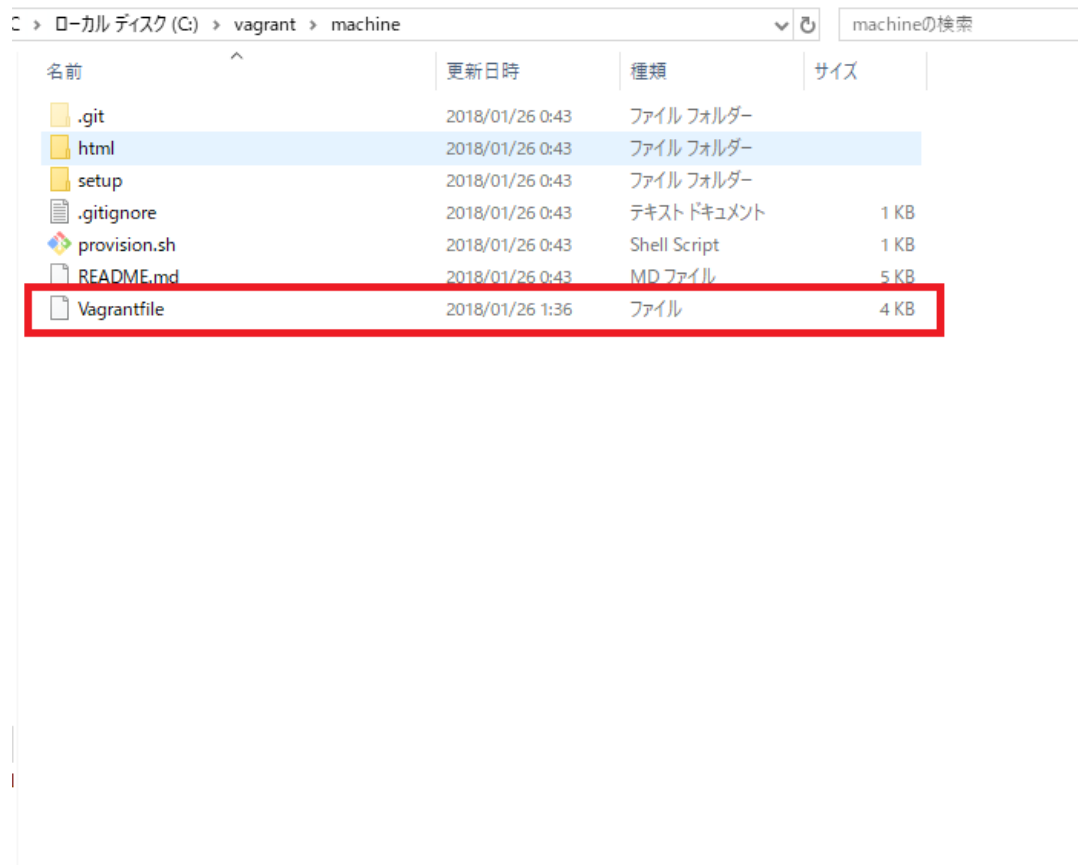


図 4.22 Vagrantfile

テキストエディタで開いたら，59 行目のメモリ容量を 2 GB 以上に増やす．

```
40 # your network.
41 # config.vm.network "public_network"
42
43 # Share an additional folder to the guest VM. The first argument is
44 # the path on the host to the actual folder. The second argument is
45 # the path on the guest to mount the folder. And the optional third
46 # argument is a set of non-required options.
47 # config.vm.synced_folder "../data", "/vagrant_data"
48 config.vm.synced_folder "../html", "/var/www/html"
49
50 # Provider-specific configuration so you can fine-tune various
51 # backing providers for Vagrant. These expose provider-specific options.
52 # Example for VirtualBox:
53
54 config.vm.provider "virtualbox" do |vb|
55   # Display the VirtualBox GUI when booting the machine
56   # vb.gui = true
57   #
58   # Customize the amount of memory on the VM:
59   vb.memory = "1024"
60   vb.cpus = 2
61 end
62
63 # View the documentation for the provider you are using for more
64 # information on available options.
65
66 # Define a Vagrant Push strategy for pushing to Atlas. Other push strategies
67 # such as FTP and Heroku are also available. See the documentation at
68 # https://docs.vagrantup.com/v2/push/atlas.html for more information.
69 # config.push.define "atlas" do |push|
70 #   push.app = "YOUR_ATLAS_USERNAME/YOUR_APPLICATION_NAME"
71 # end
72
73 # Enable provisioning with a shell script. Additional provisioners such as
74 # Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the
```

図 4.23 Vagrantfile メモリ増設

以上が終わったら，仮想マシンを起動し，ログインをする．これは 4.3.3 章に記述したとおりである．なお，初回であるため，時間がかかる．

vagrant up

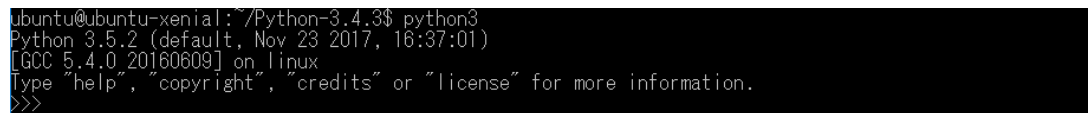
vagrant ssh

4.4.1 python の導入

```
wget https://www.python.org/ftp/python/3.4.3/Python-3.4.3.tgz
```

```
sudo apt install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-  
devel tk-devel
```

以上のコマンドを入力し実行する．そして，python と入力し，以下のような画面が出たら，Python がインストールできている．



```
ubuntu@ubuntu-xenial:~/Python-3.4.3$ python3  
Python 3.5.2 (default, Nov 23 2017, 16:37:01)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>
```


図 4.24 Python 導入

4.4.2 MeCab の導入

MeCab とは , オープンソースの形態素解析ツールである .

以下のコマンドでインストールを行こなう .

```
sudo apt install -y mecab mecab-ipadic-utf8 libmecab-dev
```



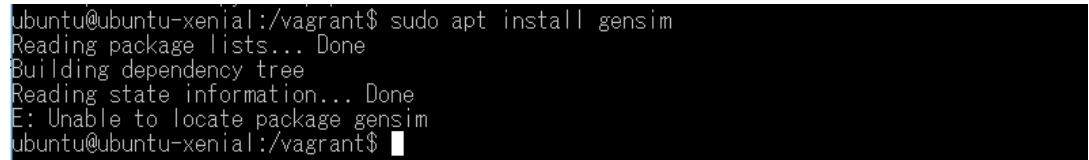
```
ubuntu@ubuntu-xenial:/vagrant$ sudo apt install -y mecab mecab-ipadic-utf8 libmecab-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package mecab
E: Unable to locate package mecab-ipadic-utf8
E: Unable to locate package libmecab-dev
```

図 4.25 MeCab 導入

4.4.3 gensim の導入

Word2vec 用ライブラリである , gensim をインストールする . 下記コマンドによってインストールできる .

```
pip install gensim
```



```
ubuntu@ubuntu-xenial:/vagrant$ sudo apt install gensim
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package gensim
ubuntu@ubuntu-xenial:/vagrant$
```

図 4.26 gensim 導入

4.4.4 コーパスダウンロード

本研究では，東北大学乾・岡崎研究所より提供されている「日本語 Wikipedia エンティティベクトル」を使用し，研究を進める．

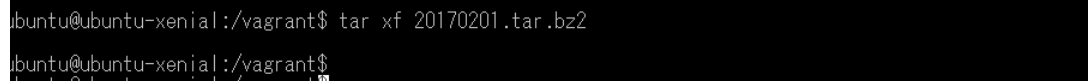
下記コマンドによって導入を行う．

```
cd /vagrant
```

```
wget http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/data/20170201.tar.bz2
```

```
tar xf 20170201.tar.bz2
```

以下のように何も出なければ成功である．



```
ubuntu@ubuntu-xenial:/vagrant$ tar xf 20170201.tar.bz2
ubuntu@ubuntu-xenial:/vagrant$
```

図 4.27 コーパス導入

4.5 Word2vec

python としてインタラクティブシェルを起動する．起動の仕方は以下の通りである．

python3 と入力する．

以下のようなになれば起動完了である．ここでは，python を利用しての作業ができる．

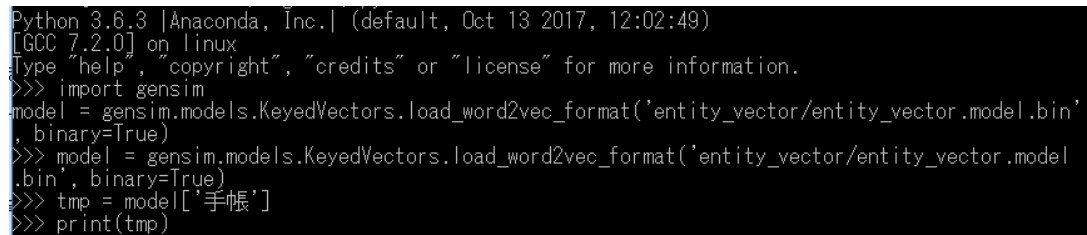
```
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

図 4.28 Python インタラクティブシェル

Word2vec は単語をベクトルで表すことができ、さらに、算出したベクトルから、単語と単語の計算をすることができる。

例えば、「手帳」という単語のベクトルを表そうとすると、下記のようなコマンドを入力する。

```
import gensim
model = gensim.models.KeyedVectors.load_word2vec_format('entity_vector/entity_vector.model.bin',
binary=True)
tmp = model['手帳']
print(tmp)
```



```
Python 3.6.3 [Anaconda, Inc.] (default, Oct 13 2017, 12:02:49)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import gensim
model = gensim.models.KeyedVectors.load_word2vec_format('entity_vector/entity_vector.model.bin',
binary=True)
>>> model = gensim.models.KeyedVectors.load_word2vec_format('entity_vector/entity_vector.model
.bin', binary=True)
>>> tmp = model['手帳']
>>> print(tmp)
```

図 4.29 「手帳」のベクトル

結果は下記のように表示される．

```
[
-1.24068546e+00  1.07822740e+00 -7.63420045e-01 -1.00119340e+00
 1.49300849e+00  4.37720381e-02 -2.78641790e-01  1.46043217e+00
-1.61940753e+00  6.96862698e-01 -8.23609769e-01  1.74395263e-01
-2.45925054e-01 -5.90076149e-01  9.07261595e-02 -8.77198339e-01
 1.22008383e+00 -2.18061781e+00 -3.62138212e-01  3.00616354e-01
 1.34302899e-01  2.29600877e-01 -1.01829076e+00 -4.54425663e-01
-8.06471825e-01 -4.60811913e-01 -8.73449028e-01 -6.02467895e-01
-1.27852738e+00  1.12160790e+00  1.05689824e-01 -1.46476686e-01
 2.41520301e-01  1.20032251e+00 -3.31092952e-03  3.88571590e-01
-6.24046028e-01 -1.61640227e+00  7.16763496e-01  1.05249667e+00
 4.53288034e-02 -1.03609729e+00 -8.17928374e-01 -1.14537477e+00
 1.25779641e+00  3.17145914e-01 -8.29142511e-01 -1.29628265e+00
 3.61937821e-01  7.55915880e-01 -3.94525439e-01  2.41006725e-02
 1.10941672e+00  6.60506487e-01  5.48565090e-01 -1.13453031e+00
-1.31629860e+00  8.45757842e-01 -1.75376981e-01  5.00822544e-01
 1.60006309e+00  5.68642169e-02  7.42048770e-02  1.19660139e+00
 8.53514895e-02 -3.37474883e-01 -2.29821515e+00 -6.10926092e-01
-1.72781516e-02 -5.98312616e-01 -3.20888311e-01 -2.08479837e-01
-7.21456230e-01 -3.40917498e-01 -1.05832314e+00  1.07937241e+00
 1.80642748e+00  2.90941417e-01 -1.54837325e-01 -5.98150313e-01
-6.14815593e-01 -1.67165682e-01  6.33179069e-01 -3.82521689e-01
 7.92609215e-01  2.13632131e+00 -4.75676537e-01  3.87118995e-01
-1.14538729e+00  7.64414549e-01 -2.58797735e-01  1.19392908e+00
 2.70826191e-01  9.31630909e-01 -5.95159650e-01 -1.10546958e+00
 6.09751761e-01 -3.61443996e-01 -1.10843611e+00  4.22547311e-01
 4.85828608e-01  5.67007363e-01 -4.88177478e-01 -4.39335071e-02
 7.89212465e-01  3.82541418e-01  1.83657968e+00  1.37353921e+00
 6.51449636e-02  2.09232286e-01 -1.00387055e-02 -1.59596741e+00
 2.63085067e-01 -1.41907561e+00 -6.33172631e-01  5.97439706e-02
 3.07916682e-02 -7.22453356e-01  2.75254697e-01  9.47807610e-01
-6.12539053e-01 -2.07103276e+00 -1.97593104e-02 -7.70888567e-01
-9.48604107e-01 -7.86301553e-01 -2.41633391e+00  9.48548019e-01
-3.50552589e-01 -6.83866441e-02 -2.24523947e-01 -1.83631631e-03
-3.03900003e-01  4.66044873e-01  5.52806258e-01  7.18727171e-01
-5.42907894e-01  1.41557485e-01  2.08435044e-01 -7.23623455e-01
 1.58955193e+00  9.74286854e-01  2.09030524e-01  4.15710807e-01
-8.08829069e-01  9.09091830e-01  2.59020060e-01  5.40854037e-01
 8.26419890e-01  1.79301226e+00  8.49199593e-01 -1.07697415e+00
-1.21219254e+00 -1.74517167e+00 -1.27662623e+00 -5.69647729e-01
-9.18444753e-01  2.96055287e-01  4.14037317e-01 -1.20909452e+00
 1.45377123e+00  8.58341515e-01 -1.60862434e+00 -1.46370006e+00
-7.77853504e-02  2.98508078e-01  6.93228185e-01 -6.61946893e-01
 1.21265493e-01 -1.37738302e-01  8.98371458e-01  4.10278827e-01
-6.11464381e-01  7.01255739e-01  6.68690145e-01 -6.92157924e-01
-4.79012400e-01  2.96447068e-01 -5.63345253e-02  1.68011582e+00
 8.97411287e-01  2.86374683e-03  5.85057557e-01  1.01569705e-01
-6.47467732e-01  8.15762877e-01  1.34934998e+00  1.09976006e+00
 5.99752545e-01  1.66144288e+00 -5.79154611e-01  8.52900624e-01
-5.03976978e-02 -2.64479131e-01 -2.12308794e-01  1.38052687e-01
-1.60443437e+00  2.59145290e-01 -1.10608757e+00  1.20742643e+00]
```

図 4.30 「手帳」のベクトル 結果

4.5.1 R の導入

R 言語とは、統計解析やその結果をグラフィカルに表示するためのシステム「R」用の言語のことである。R 言語は、AT&T ベル研究所の研究者によって設計された統計処理言語である S 言語を元に設計されている。同じく AT&T ベル研究所のが開発した「S 言語」の実装系は商用版が知られているが、R 言語は GNU プロジェクトによってオープンソースで提供されており、無償で利用することができる。R 言語は、簡単なコマンドによりいろいろな機能が実現できる。標準では用意されていない機能も比較的容易に拡張できるメリットがある。

インストールは、chocolatey から行うことができる。

windows の管理者権限のあるコマンドプロンプトを立ち上げ、以下のコマンドを入力する。

```
choco install r.project
```



図 4.31 R のインストール

次に、実際に本研究と同内容の word2vec の扱いをしていく。大まかな流れとしては以下の通りである。

1. 1 文章ごとに区切る。
2. タグ付けを行い、CSV ファイルにまとめる。
3. MeCab を利用し、分かち書き処理をさせる。
4. ベクトル化する。
5. 主成分分析を行う。
6. 比較・考察を行う。

4.5.2 解析対象文

私自身の 3 年次の課題研究概要で書いた文章を例に手法を書く．解析対象文は以下のとおりである．

少子高齢化が進み，健康寿命が短くなっている現在，介護業界はこれから重要となり，需要も増加傾向にある業界である．実際に特別養護老人ホームの入所申込者数（待機者数）は 09 年～14 年の 5 年間で 10 万人増加している．待機者数増加の要因として考えられるのは，1947 年～1949 年に生まれた団塊世代の人たちが徐々に介護サービスを必要としてきているからである．

介護職員は賃金，労働時間，体力的，精神的な負担が大きい．これらの要因から介護現場は厳しい労働環境であり，離職率が高い．さらに平成 26 年時点で介護分野における有効求人倍率が 2 倍を超えている状況であるため，増える介護の需要に介護職員の人数が追いついていないといえる．介護現場の人材不足を解消するには介護のオートメーション化や外国人労働者の雇用，介護職員の負担軽減が必要であると考えられる．

以上の 2 段落 7 文章である．

この文章を解析用ファイルとして下図のように CSV ファイルにする。
タグ A は 1 段落目を表し、タグ B は 2 段落目を表している。
これを、仮に " 01.csv " として保存する。

1	A、少子高齢化が進み、健康寿命が短くなっている現在、介護業界はこれから重要となり、需要も増加傾向にある業界である。
2	A、実際に特別養護老人ホームの入所申込者数(待機者数)は09年～14年の5年間で10万人増加している。
3	A、待機者数増加の要因として考えられるのは、1947年～1949年に生まれた団塊世代の人たちが徐々に介護サービスを必要としてきているからである。
4	B、介護職員は賃金、労働時間、体力的、精神的な負担が大きい。
5	B、これらの要因から介護現場は厳しい労働環境であり、離職率が高い。
6	B、さらに平成26年時点で介護分野における有効求人倍率が2倍を超えている状況であるため、増える介護の需要に介護職員の人数が追いついていないといえる。
7	B、介護現場の人材不足を解消するには介護のオートメーション化や外国人労働者の雇用、介護職員の負担軽減が必要であると考えられる。
8	

図 4.32 解析前 CSV ファイル

保存が完了したら，vagrant 上で分かち書き処理をし，ベクトル化処理をする．

下図のように何もエラーがでなければ成功である．

```
cat 01.csv | python s2v.py > 01vec.csv
```

A terminal window titled 'コマンドプロンプト - vagrant ssh' with standard window controls. The terminal shows the command 'cat 01.csv | python s2v.py > 01vec.csv' being executed. The prompt changes from 'ubuntu@ubuntu-xenial:~\$' to 'ubuntu@ubuntu-xenial:/vagrant\$' after the command is run. The rest of the terminal area is black, indicating no further output or errors.

```
ubuntu@ubuntu-xenial:~$ cat 01.csv | python s2v.py > 01vec.csv
ubuntu@ubuntu-xenial:/vagrant$
```

図 4.33 ベクトル化

ベクトル化されたデータは , " 01vec . csv " として同フォルダ内に保存されている .

名前	更新日時	種類	サイズ
.git	2018/01/12 7:46	ファイル フォルダー	
.vagrant	2018/01/12 7:47	ファイル フォルダー	
entity_vector	2017/02/17 23:29	ファイル フォルダー	
html	2018/01/12 7:46	ファイル フォルダー	
setup	2018/01/12 7:46	ファイル フォルダー	
.gitignore	2018/01/12 7:46	テキストドキュメント	1 KI
01.csv	2018/01/26 9:29	Microsoft Excel C...	2 KI
01vec.csv	2018/01/26 9:29	Microsoft Excel C...	21 KI
20170201.tar.bz2	2017/02/18 0:22	BZ2 ファイル	1,341,398 KI
Anaconda3-5.0.1-Linux-x86_64.sh	2018/01/12 8:05	Shell Script	537,888 KI
provision.sh	2018/01/12 7:46	Shell Script	1 KI
README.md	2018/01/12 7:46	MD ファイル	5 KI
s2v.py	2017/12/22 10:29	Python File	1 KI
ubuntu-xenial-16.04-cloudimg-console.l...	2018/01/26 9:00	テキストドキュメント	40 KI
Vagrantfile	2018/01/12 8:50	ファイル	4 KI

図 4.34 ベクトルされたデータ

ベクトル化され出力されたデータの中身はこうになっている．次にこの数値を主成分分析へかけていく．

```
1 A,-0.0943072587252,-0.00359934708104,-0.587403833866,-0.381105393171,1.59847033024,1.54584920406,-0.132757529616,0.335498660803,  
2 A,-1.70123839378,0.505578577518,-1.7803208828,0.222575768828,-0.156001999974,1.47881138325,0.895097255707,0.929757595062,0.56588  
3 A,0.834630489349,-1.2264316082,-1.61906814575,-1.3182195425,0.541914284229,-0.709017574787,-0.748380303383,0.544743359089,-1.433  
4 B,0.0604689307511,-0.692482769489,-1.59988439083,-2.20076560974,1.37935113907,0.670099318027,-0.0770919620991,1.98370075226,-1.1  
5 B,-1.51588845253,0.416150152683,-1.38416111469,-0.339009255171,-0.431858718395,0.21779447794,1.94928956032,0.0645260736346,-1.2  
6 B,-0.714731693268,-0.605653941631,-2.02402472496,-1.37816619873,0.955163836479,-0.723249852657,-0.298135310411,1.87637412548,-1.  
7 B,0.0604689307511,-0.692482769489,-1.59988439083,-2.20076560974,1.37935113907,0.670099318027,-0.0770919620991,1.98370075226,-1.1  
8
```

図 4.35 01vec.csv

R を起動し、きれいな描画のためのツールをインストールする。

コマンドは下記の通りである。

```
install.packages( " devtools " )
```

```
devtools::install_github( " vqv/ggbiplot " )
```

下図のようにエラーが出なければインストール完了である。



```
R Console

R version 3.4.2 (2017-09-28) -- "Short Summer"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してください。

R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' と入力してください。
また、R や R のパッケージを出版物で引用する際の形式については
'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます。
'q()' と入力すれば R を終了します。

> install.packages('devtools')
> devtools::install_github('vqv/ggbiplot')
> |
```

図 4.36 きれいな描画のためのツール

続いて、主成分分析へ入る．

```
setwd('c:/vagrant/machine')#作業ディレクトリの変更
```

```
myData <- read.csv('01vec.csv', head = F)
```

```
myResult <- prcomp(myData[, -1])#主成分分析
```

```
library(ggbiplot)
```

```
ggbiplot(myResult, var.axes = F, groups = myData[, 1])
```

このコマンドを実行し，エラーが出なければ成功である．

```
>
>
>
>
> setwd('c:/vagrant2/machine')
> myData <- read.csv('01vec.csv', head = F)
> myResult <- prcomp(myData[, -1])
>
> library(ggbiplot)
要求されたパッケージ ggplot2 をロード中です
要求されたパッケージ plyr をロード中です
要求されたパッケージ scales をロード中です
要求されたパッケージ grid をロード中です
警告メッセージ:
1: パッケージ 'ggplot2' はバージョン 3.4.3 の R の下で造られました
2: パッケージ 'plyr' はバージョン 3.4.3 の R の下で造られました
3: パッケージ 'scales' はバージョン 3.4.3 の R の下で造られました
> ggbiplot(myResult, var.axes = F, groups = myData[, 1])
>
>
> |
```

図 4.37 R で主成分分析

主成分分析の結果である． これを比較し，考察していく．

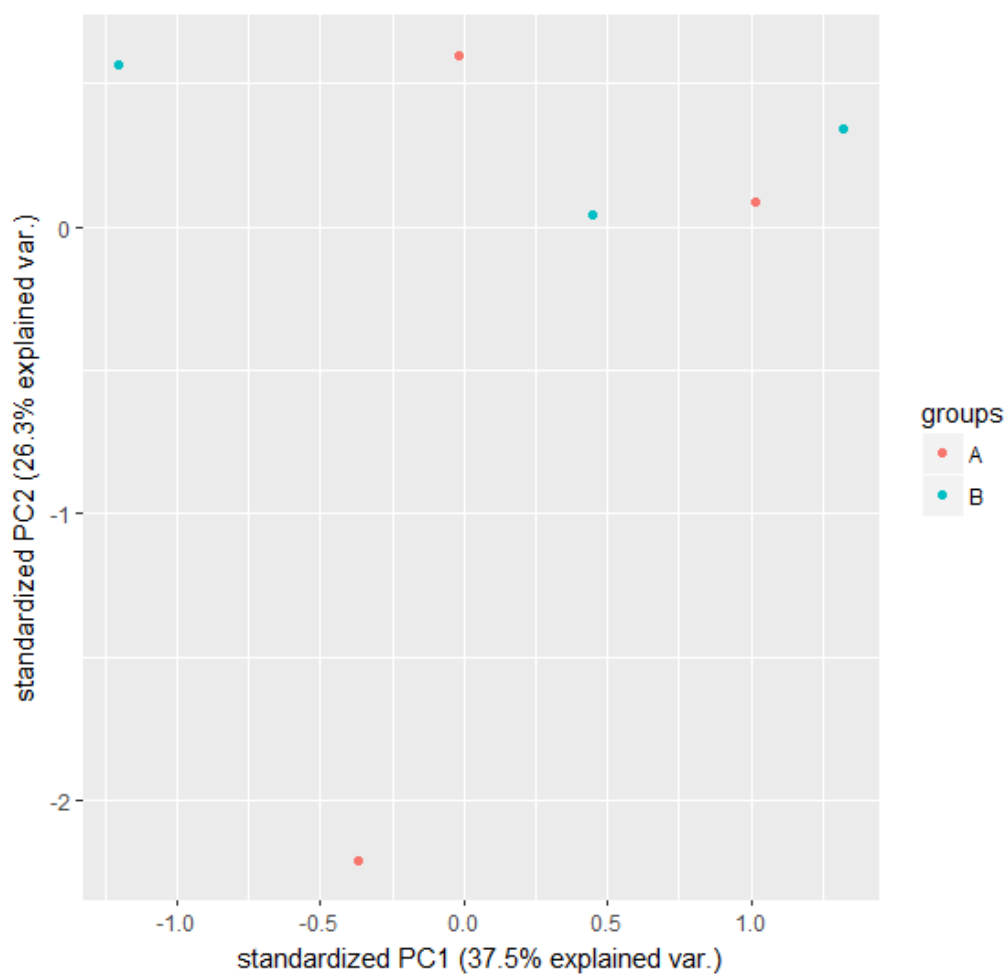


図 4.38 主成分分析の結果

第 5 章

結果

word2vec を用い , 数値的に文章構造の解析をした結果である . 以下が解析対象文である .

少子高齢化が進み , 健康寿命が短くなっている現在 , 介護業界はこれから重要となり , 需要も増加傾向にある業界である . 実際に特別養護老人ホームの入所申込者数 (待機者数) は 09 年 ~ 14 年の 5 年間で 10 万人増加している . 待機者数増加の要因として考えられるのは , 1947 年 ~ 1949 年に生まれた団塊世代の人たちが徐々に介護サービスを必要としてきているからである .

介護職員は賃金 , 労働時間 , 体力的 , 精神的な負担が大きい . これらの要因から介護現場は厳しい労働環境であり , 離職率が高い . さらに平成 26 年時点で介護分野における有効求人倍率が 2 倍を超えている状況であるため , 増える介護の需要に介護職員の人数が追いついていないといえる . 介護現場の人材不足を解消するには介護のオートメーション化や外国人労働者の雇用 , 介護職員の負担軽減が必要であると考えられる .

ここまで解析対象文である .

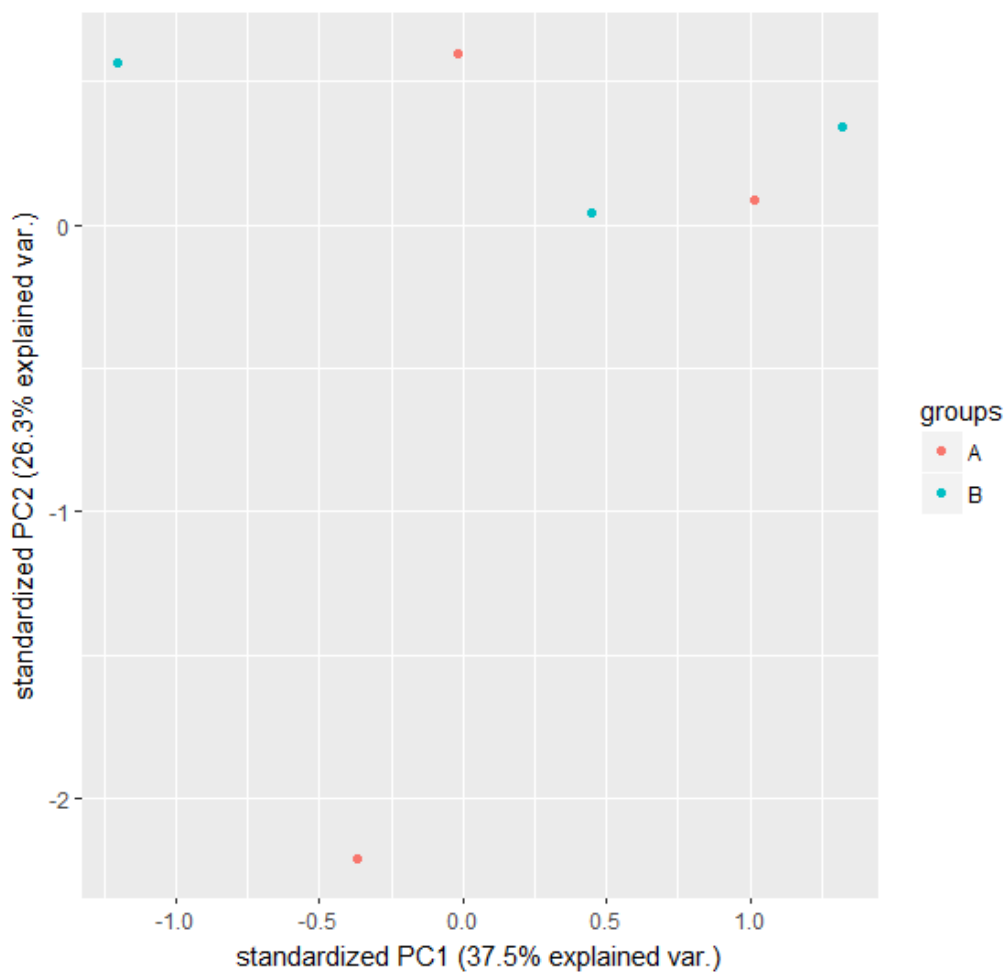


図 5.1 主成分分析の結果

2 つめの解析対象文章である．これは新聞記事である．

総務省が 26 日発表した 2017 年 12 月の全国消費者物価指数（C P I、2015 年 = 100）は、値動きの大きな生鮮食品を除く総合指数が 100.7 と、前年同月比 0.9 % 上昇した。プラスは 12 カ月連続。Q U I C K がまとめた市場予想の中央値（0.9 % 上昇）と同水準だった。ガソリンなどエネルギー価格上昇の影響が大きかった。

生鮮食品を含む総合は 101.2 と 1.0 % 上昇した。エネルギー価格上昇のほか、レタスなど葉物野菜の生育遅れと、ピールの値上がりなども押し上げ要因だった。一方で携帯電話料金や家電価格は下落しており、生鮮食品とエネルギーを除く総合は 101.0 と、0.3 % の上昇にとどまった。

出典：2018/1/26 日本経済新聞

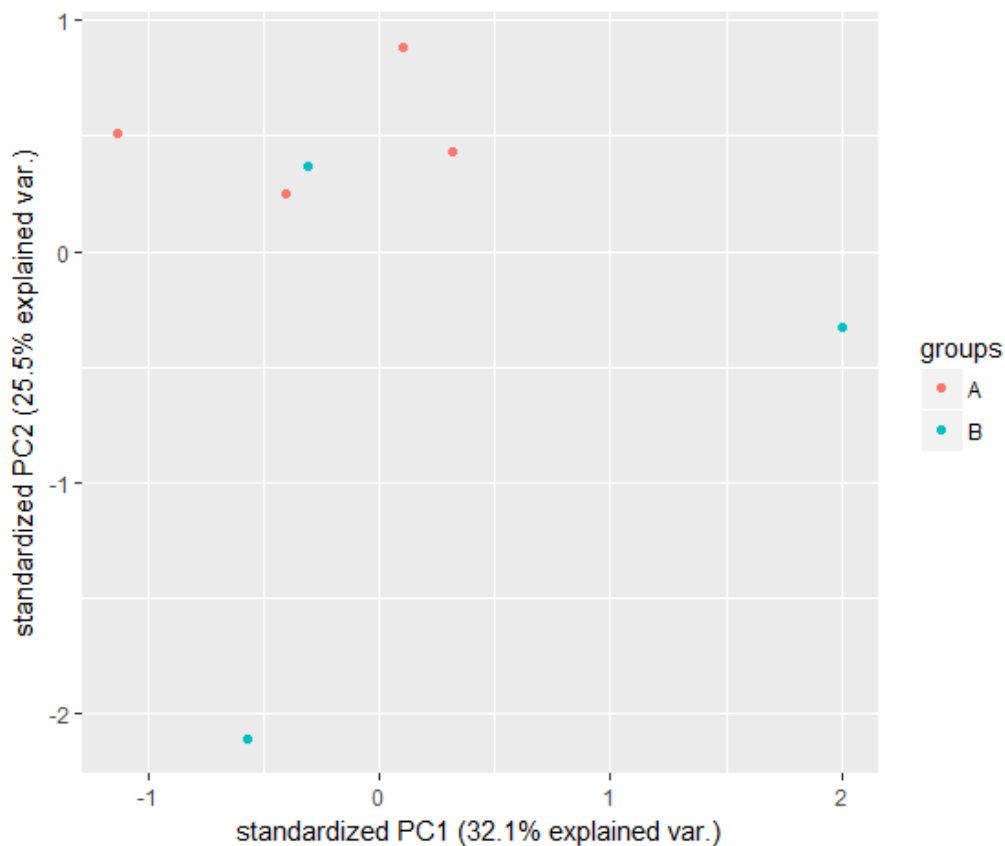


図 5.2 分析結果

以下が 3 つ目の解析対象文である。

内閣府は 23 日開いた経済財政諮問会議（議長・安倍晋三首相）で、「中長期の経済財政に関する試算」を提出した。試算では高成長シナリオで実質成長率が 2020 年度に 1.5 %、20 年代前半から 2 %程度に達すると想定。国と地方の基礎的財政収支（プライマリーバランス、P B）の黒字化は昨年 7 月の試算より 2 年遅れて 27 年度になるとみている。

内閣府は毎年、年初と夏に今後 10 年の成長率と財政の姿をまとめた中長期試算を公表する。今回は実質成長率が 20 年代前半に 2 %に達する「成長実現ケース」と、1 %強の成長が続く「ベースラインケース」の 2 通りを示した。これまでも 2 通りのシナリオを公表してきたが、今回から高成長シナリオを過去の政策効果の実績を踏まえた現実的なものに変更した。 出典：2018/1/23 日本経済新聞

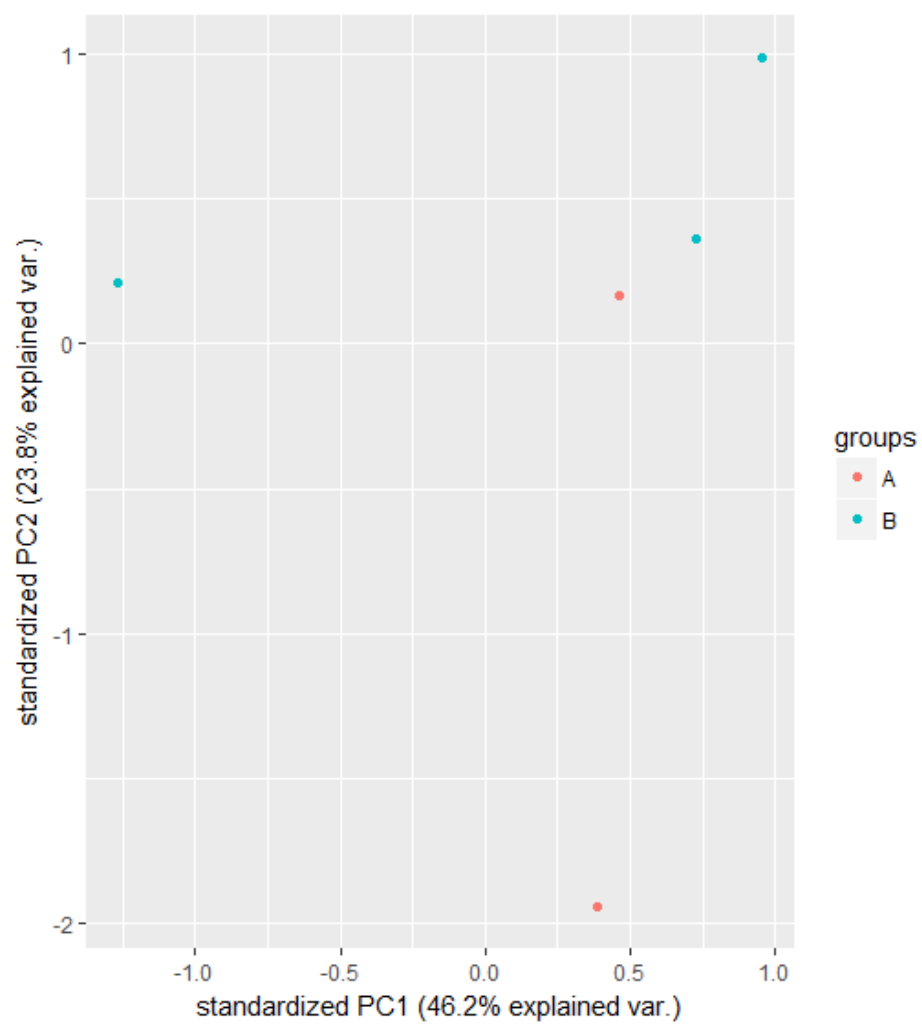


圖 5.3 分析結果

第 6 章

考察

このグラフからは，数値が散らばっているので，段落内で同じ話題が書かれているとは限らない．と考える事ができる．

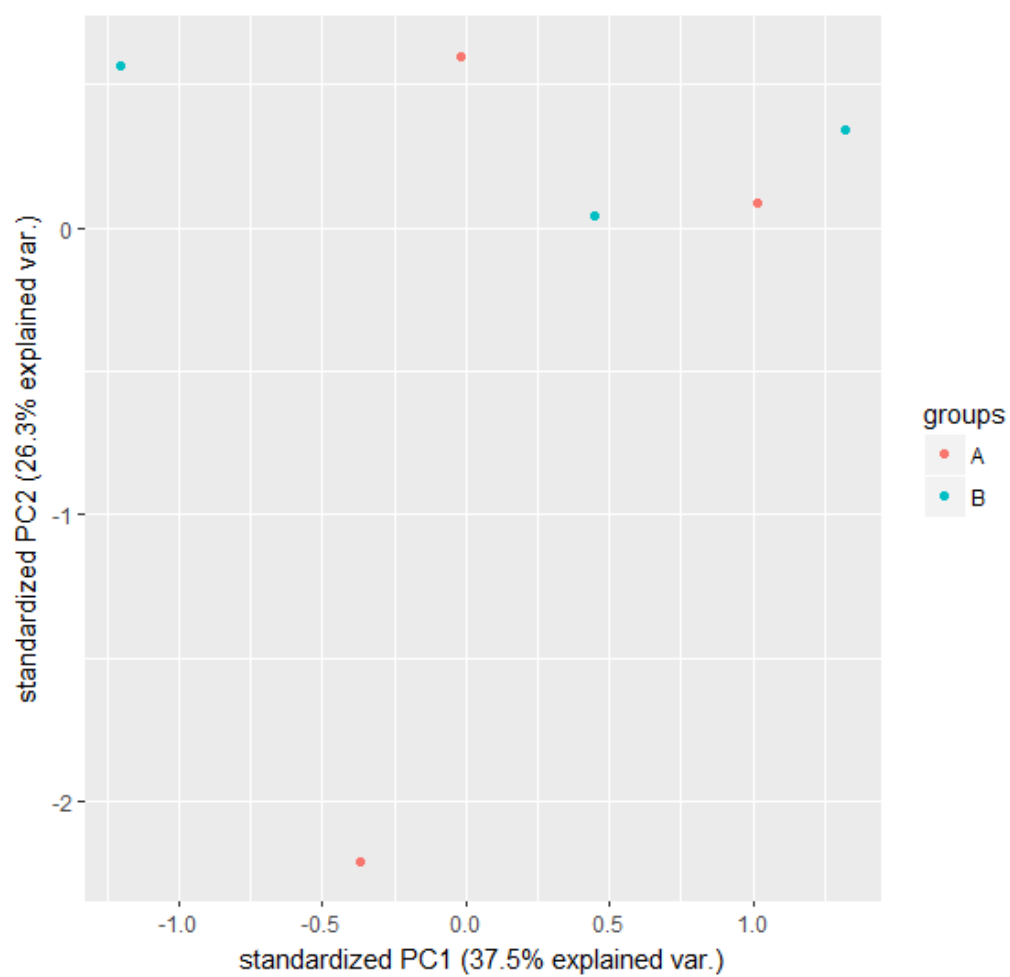


図 6.1 主成分分析の結果

タグ A の数値は割りと近い位置にプロットされており，話題の方向性が近いといえる．
しかし，タグ B はバラバラに散らばっているので方向性が近いとは言い難い．

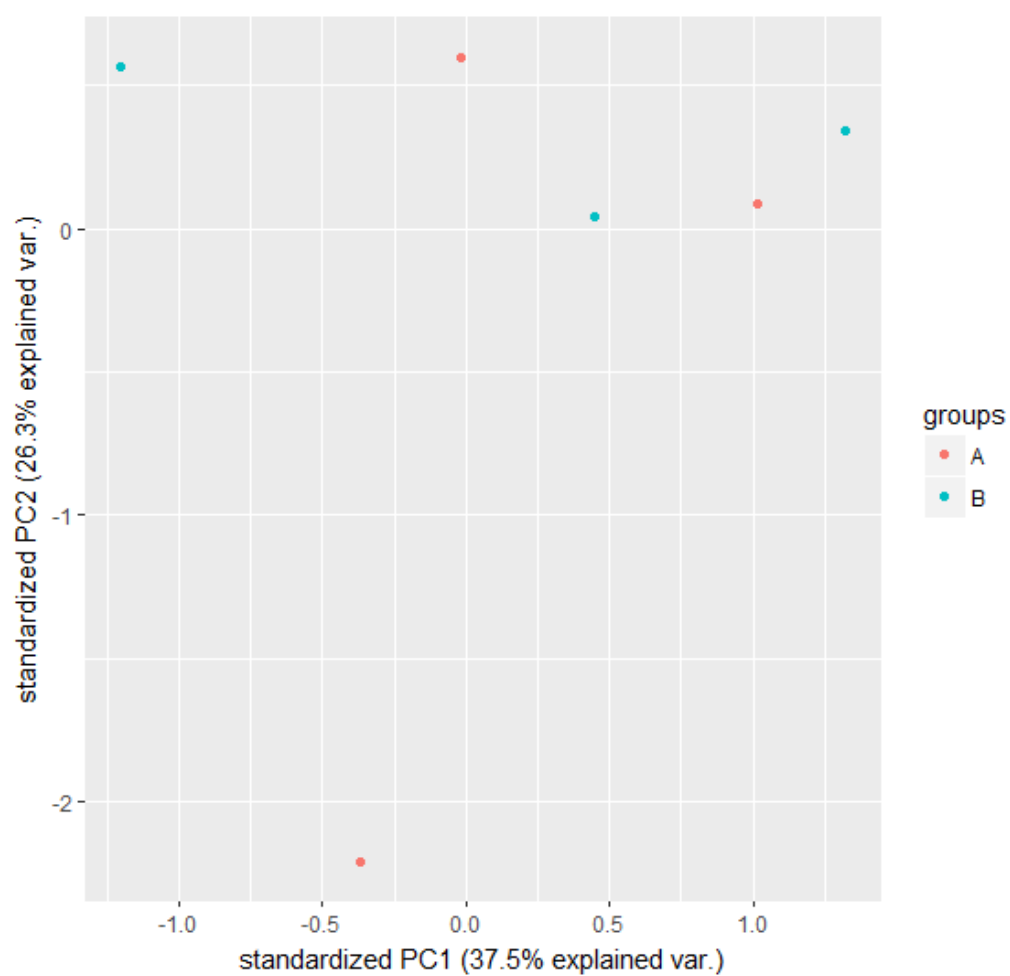


図 6.2 主成分分析の結果

全体的に散らばっており，同じ話題の方向性とは言い難い．

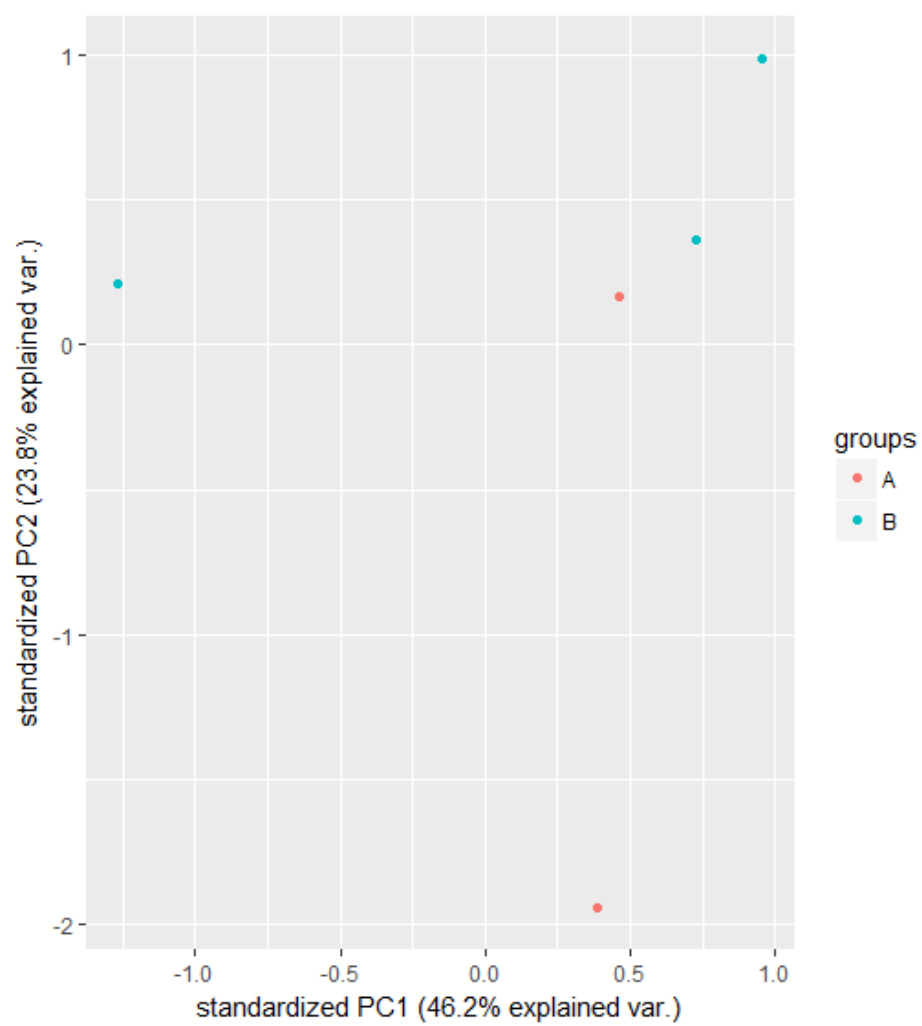


図 6.3 分析結果

同段落内の文章は同じ話題でなければならぬため、文章のベクトルも同じ方向性である必要がある。

分析結果では、一文章ごとの数値がグラフにプロットされている。このことから、文章の方向性が同じならば、タグ A とタグ B に対応する点がそれぞれ別々に集まることが期待される。

本研究での分析結果は、同段落内の文章にもかかわらず、それぞれのタグに対応する点が散らばって分布している。従って、解析対象文章の話題の方向性はバラバラであったと考えられる。

第 7 章

結論

今回の研究から，Word2vec を用いてベクトルへ変換した文章を定量的に検証することで，個人の主観による添削だけでなく，定量的な文章の添削を行うことが期待される．

参考文献

- [1] 西尾泰和. word2vec による自然言語処理. 株式会社オライリー・ジャパン, 第 2 版, 2017.
- [2] 倉島保美. 論理が伝わる 世界標準の「書く技術」. 講談社, 2012.
- [3] 日本経済新聞. 基礎的財政収支、2027 年度に黒字化 諮問会議試算. https://www.nikkei.com/article/DGXLASFL23HBS_T20C18A1000000/?n_cid=SPTMG053 (2018.01.24 閲覧).
- [4] 日本経済新聞. 17 年の全国消費者物価 0.5 %上昇 12 月は 0.9 %上昇. https://www.nikkei.com/article/DGXLASFL26H78_W8A120C1000000/?nf=1 (2018.01.26 閲覧).

謝辞

本研究を進めるにあたり，矢吹太郎准教授を始めとした矢吹研究室の先輩，同期，後輩の皆様には多くの時間を割いて頂き，様々なご助言をいただきました．この場を借りて深く御礼申し上げます．