テキストマイニングの実践
― 2日目 ―

2021/7/2

人文社会ビジネス科学学術院
ビジネス科学研究群

講義スライド

- https://github.com/haradatm/lecture/tree/master/**gssm-202107**

(参考) サンプルコード

※ 講義範囲を超えるため,解説は行いませんが Python スクリプトを利用したデータの取得や解析の参考に紹介します

- https://github.com/haradatm/lecture/tree/master/**gssm-202107**

master　lecture / gssm-202107 /

Tomohiko HARADA and Tomohiko HARADA

..

- 00-slides
- 01-data
- 02-tools
- 03-samples
- 05-colab
- README.md

README.md

Sample code for Google Colaboratory

Open in Colab

Scripts for making datasets (not used in the course)

file name	memo
prepare_rakuten_dataset.ipynb	Pre-process for rakuten dataset
prepare_covid19_dataset.ipynb	Pre-process for covid19 dataset

More sample scripts (not used in the course)

file name	memo
bbs_example.ipynb	**Fetch** from a BBS by scraping (using **BeautifISoup4**)
twitter_example.ipynb	**Fetch** and **analysis** from twitter (using **Twitter API**)

自然言語処理のトレンド

お話すること

- **深層学習の発展とともに自然言語処理も進化**
  - 応用タスクの学習データで End-to-end で学習可能
  - ブラックボックスのため, 出力の解釈が難しい等の課題もある
- **自然言語処理研究の界隈では, BERT (Transformer) が席巻中**
  - テキスト分類, 機械翻訳, 質問応答(機械読解), 要約, 文生成 など様々な応用タスクの性能を向上
  - BERT 自体のブラックボックスを解明する “Bertology” も盛ん

NEW

お話すること (2021)

- **深層学習の発展とともに自然言語処理も進化**
  - 応用タスクの学習データで End-to-end で学習可能
  - ブラックボックスのため, 出力の解釈が難しい等の課題もある
- **自然言語処理研究を超えて Transformer が席巻中**
  - BERT (Transformer) テキスト分類, 機械翻訳, 質問応答(機械読解), 要約, 文生成 など様々な自然言語処理の応用タスクの性能を向上
  - GPT-3 などで使われている Transformer が, 画像認識など自然言語処理以外の領域でも成果を発揮しつつある
- **トレンドは 視覚+言語の事前学習モデル** で,研究が盛ん

形態素解析器

出所: https://taku910.github.io/mecab/ をもとに作成

形態素解析器	ChaSen	MeCab	JUMAN	JUMAN++
コスト推定	HMM	CRF	人手	**RNNLM**
探索方法	接続コスト最小法 (ビタビアルゴリズム)			
連携する構文解析器	~~Cabocha~~	Cabocha	KNP	

深層学習を使った手法で, **自然な言葉の繋がり**を考慮

単語分割+品詞タグ付け精度 (F1)

処理速度 (文/秒)

学習・評価データ

京都大学テキストコーパス (NEWS),
京都大学ウェブ文書リードコーパス (WEB)

RNN言語モデルの学習

Webコーパス 1000万文

出所:
https://drive.google.com/file/d/1DVnrsWw4skRgC8jU6_RkeofOQEHFwctc/view?usp=sharing

(参考) 形態素解析の辞書

辞書	JUMAN辞書	Ipadic (NAIST-jdic)	UniDic	NEologd
コーパス	京都大学テキスト コーパス	RWCコーパス*2	BCCWJ コアデータ*3	RWCコーパス
形態素解析器	JUMAN MeCab	Chasen MeCab	MeCab	MeCab
単語長	長い	やや短い	短い	とても長い

*1 毎日新聞 1995年の記事や社説 4万文
*2 旧通産省主導のプロジェクト,毎日新聞1994年3000記事 約3万7千文(約91万語)
*3 現代日本語書き言葉均衡コーパス, 国語研が中心となり開発, 書籍,雑誌,新聞,Webなど, 9万単語

自然言語処理

- 機械翻訳

- 検索

- AIアシスタント

自然言語をコンピュータで処理するための技術

自然言語処理

- 基礎タスク
  - 言語を応用タスクで利用しやすい形式に変換する
    例: 形態素解析(品詞タグ付け), 構文解析, 意味解析など

- 応用タスク
  - 自然言語処理を応用したアプリケーション
    例: 機械翻訳, 質問応答, テキスト要約, 対話システムなど

自然言語処理タスクの例

基礎タスク	形態素解析 (品詞タグ付け)	文をそれぞれの意味を担う最小の単位(＝形態素)に分割し,それぞれに品詞などの情報を付与する (例: MeCab, JUMAN++)
	構文解析	形態素解析で分割した単語同士の関連性を解析し,主に文節間の係り受け構造を発見しツリー化する, 文中の単語間の係り受け関係を調べ,どの単語がどの単語に係るのかを構文的に解析する係り受け解析 (例: CaboCha, KNP, SpaCy) や, 語および文法的カテゴリを節点とするツリー形式によって文の構造を表現した句構造解析 (例: Stanford Core NLP) がある
	意味解析	与えられた文のを明らかにする処理は何でも意味解析と呼ばれる, 格解析, 述語項構造解析, 多義性解消, 比喩理解 などが例として挙げられる
応用タスク	機械翻訳	自然言語によるある言語の文を入力とし,これを違う言語の文に翻訳する
	質問応答	自然言語による質問文を入力として受け取り,適切な回答を返す
	テキスト要約	与えられた文章を短く簡潔にまとめる, 文章の一部を抜粋して要約を作成する抽出型要約と,元の文章に存在しない文章で要約を作成する抽象型要約がある
	対話システム	自然言語により人間と機械が対話を行う,チャットボットなどに使用されている

トレンドは, 深層学習の導入

- 大規模な訓練データで応用タスク全体を学習 → End-to-end 学習

坪井,海野,鈴木. 深層学習による自然言語処理. 講談社, 2017, p.4 の図を一部修正

深層学習 (ディープラーニング)

- **ニューラルネット**を用いた機械学習手法
  - 機械学習とは,データを学習し,パラメータを獲得すること
  - 脳の神経細胞(ニューロン)の働きを模した仕組みや構造のこと

ニューラルネットの歴史

- 黎明～終焉を繰り返し,近年は 3度目のブーム

第1期	1940～	• McCullochとPitts が形式ニューロンモデルを発表 [McCulloch-Pitts,43]
	1950～	• Rosenblatt がパーセプトロンを発表 [Rosenblatt,57]
	1960～	• MinskyとPapert が単純パーセプトロンの(線形分離不可能問題への)限界を指摘 [Minsky-Papert,69]
冬	1970～	冬の時代 (階層的構造の学習方法が未解決)
第2期	1980～	• Fukushima らがネオコグニトロンを提案 [Fukushima,80] • Rumelhart らが誤差逆伝播法を提案 [Rumelhart+,86] • LeCun らが畳み込みニューラルネット Conv.net を提案 [LeCun,89]
冬	1990～	冬の時代 (学習時間や過学習に課題, 一方でSVMが流行)
第3期	2000～	• Hinton らが事前学習とオートエンコーダを導入した多層NNを提案 [Hinton+,06]
	2010～	• Seide らが音声認識のベンチマークで圧勝 [Seide+,11] • Krizhevsky らがReLU を提案し画像認識コンペで圧勝 [Krizhevsky,12]

様々なニューラルネット

Feed forward NN

CNN (畳み込みNN)

RNN (Recurrent NN)

Recursive NN

AutoEncoder

音声認識で成功 [Seide+,2011]

- Microsoft Research のグループ
  - 電話での会話音声の標準データセット
  - 入力(MFCC)-出力(HMM状態変数)の関係をDNNで学習
    - 従来 GMM-HMM → DNN-HMM (全結合7層, 事前学習あり)
  - 単語誤認識率で 10%前後の大幅な精度改善

acoustic model & training	recognition mode	RT03S		Hub5'00	voicemails		tele-
		FSH	SW	SWB	MS	LDC	conf
GMM 40-mix, ML, SWB 309h	single-pass SI	30.2	40.9	26.5	45.0	33.5	35.2
GMM 40-mix, BMMI, SWB 309h	single-pass SI	27.4	37.6	23.6	42.4	30.8	33.9
CD-DNN 7 layers x 2048, SWB 309h, this paper	single-pass SI	18.5	27.5	16.1	32.9	22.9	24.4
(rel. change GMM BMMI → CD-DNN)		(-33%)	(-27%)	(-32%)	(-22%)	(-26%)	(-28%)

F. Seide, G. Li and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." *Interspeech*. 2011.

画像認識で成功 [Krizhevsky+,2012]

- 一般物体認識 (Hintonのグループ)
  - ImageNet Large-scale Visual Recognition Challenge 2012
    - 1000カテゴリ×約1000枚 = 100万枚 の訓練画像
    - 畳込み層5, 全結合層3, 2つのGPUで2週間 (AlexNet)
    - 誤識別率が10%以上減少 (過去数年間での向上は1～2%)

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012. http://image-net.org/challenges/LSVRC/2012/supervision.pdf

画像認識における認識精度の変遷

- 2015年,人の認識精度(5.1%)を超えたことが話題に

深層学習 成功の背景

- 一定以上の規模のデータ →改善
  - WebやIoT(センサ)などから十分な規模のデータを収集可能
- 学習の難しさ → 改善
  - 様々なテクニック (事前学習, dropout 等)
- 誤差逆伝搬法の計算量膨大 → 改善
  - 計算機能力の飛躍的向上
  - GPU, マルチコアCPU, PCクラスタの登場
- 性能を引き出すのに必要なノウハウ → 未解決
  - 「黒魔術」のまま → **Explanable AI (説明可能AI)として研究が盛ん**

単語の分散表現 (=単語埋め込み)

- 単語のベクトル化 →大規模コーパスによる事前学習

既存手法	近年→事前学習モデル
TF-IDF, Okapi BM25 など (分布的, 高次元, スパース)	**word2vec**, GloVe, fastText など (分散的, 低次元, 密)

- 代表格は「word2vec」
  - 深層学習による分布仮説のモデル化
  - king – man + woman = queen で有名
    →右の例では, 日本- 東京 + ローマ = イタリア

Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig, 2013, NAACL

事前学習モデルのタイムライン

https://towardsdatascience.com/2019-year-of-bert-and-transformer-f200b53d05b9

深層学習の適用

分類	タスクの例	既存手法	深層学習による手法
基礎タスク	言語モデル	• N-gram	• Recurrent NN (RNNLM)
	分散表現	• TF-IDF • Okapi BM25	• word2vec → **BERT**
	品詞タグ付け	• CRF • SVM	• Encoder-Decoder → **BERT** ※ Seq2Seq や Attention機構を含む
応用タスク	文書分類	• TF-IDF • Okapi BM25	• Recurrent NN ※前の語を考慮 • Recursive NN ※木構造を考慮 • Convolutional NN ※付近の語を考慮 } → **BERT**
	機械翻訳	• 統計的機械翻訳	• Encoder-Decoder → **Transformer** ※ 対訳コーパスを end-to-end で学習する
	文書要約	• SVM • 最大被覆問題	• Encoder-Decoder → **Transformer** ※ 原文と要約文を end-to-end で学習する

Transformer 以前

- ニューラル機械翻訳の基本となったモデル
  - Seq2Seq [Sutskever+, NIPS2014]

“ABC”という単語列から”WXYZ”という単語列への翻訳

Transformer [Vaswani+,2017]

- RNNやCNNを使わずアテンションのみ使用した Transformer がニューラル機械翻訳で圧倒的な SOTA を達成
  - 従来,単語系列の文脈理解は主にLSTM →長期依存性の理解に限界
  - 離れた単語の関係性も直接考慮できる Self-Attention が性能向上に大きく寄与した (しかも省メモリで計算可)

Transformer [Vaswani+,2017]

- 例: レイヤーN=6, ヘッドh=8, 長さ=512, 中間層=768

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_\mathrm{h})W^O$$

Figure 1: The Transformer - model architecture.

BERT [Devlin+,2018]

- 双方向 Transformer ブロックを24層重ねた言語モデル
- 事前学習モデルが公開

- 英語
  - 本家 Google の事前学習モデル *1
  - Book Corpus 8億語 + 英語 Wikipedia 25億語 (語彙数 3万)
- 日本語
  - 黒橋研の事前学習モデル *2
  - 日本語Wikipedia 約1,800万文 (語彙数 3.2万)

*1 https://github.com/google-research/bert
*2 http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT日本語Pretrainedモデル

2018年10月: BERT の衝撃

- タスクに特化した構造を持たずに,人間のスコアを大きく超えた

SQuAD1.1 Leaderboard

Since the release of SQuAD1.0, the community has made rapid progress, with the best models now rivaling human performance on the task. Here are the ExactMatch (EM) and F1 scores evaluated on the test set of SQuAD v1.1.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304 人間	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433** BERT	**93.160**
2 Oct 05, 2018	BERT (single model) *Google AI Language* https://arxiv.org/abs/1810.04805	85.083	91.835
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202

https://rajpurkar.github.io/SQuAD-explorer/

- 機械読解タスク(左)で,完全一致と部分一致の両指標で最高精度(2018/10/5)
- 様々な自然言語理解タスクでSOTA (QA,含意,言い換え,NER等)
- タスク適応は,出力層をタスク毎に 1層のみ追加してfine-tuning

文脈を考慮した表現

- 文脈を考慮することで,様々なタスクの性能が向上

文脈に関係なく **一つの単語には一つのベクトル**が割り当てられる	周りの文脈によって **同じ単語でも異なるベクトル**が割り当てられる
首を痛める 首 会社を首になる ↕ 同じ 首	首を痛める 首 会社を首になる ↕ 異なる 首

NEW

BERT Pre-train モデル

公開元	Google Research	京大 黒橋・河原・村脇研	NICT	東北大 乾・鈴木研
日/英	英語	日本語	日本語	日本語
コーパス	14GB (Book Corpus, Wikipedia)	3GB (Wikipedia)	3GB (Wikipedia)	3GB (Wikipedia)
単語数	30K (BPE)	32K (JUMAN & BPE)	32K (MeCab+JUMAN & BPE)	32K (MeCab+Neologd & BPE)
入力長 *1	最大512トークン	最大128トークン	最大512トークン	最大512トークン
パラメータ	24層, 各層1024次元	24層, 各層1024次元	12層, 各層768次元	12層, 各層768次元
学習時間	16Cloud TPUs で 4日間(≒100時間)	1GPU (GTX 1080 Ti) で 約30日間(≒750時間)*2	32GPU (V100) で約 7 日間(≒175時間)	8Cloud TPUs で 約14日間(≒350時間)

*1 入力できるシーケンスの長さに制限があることに注意

*2 表中のパラメタは LARGEモデル, 学習時間のみ BASEモデル(12層, 768次元)の場合

HuggingFace's **Transformers**

https://huggingface.co/

- Huggingface が提供する Pytorch によるフレームワーク
- 簡単にBERTなどの汎用言語モデルを動かせる

README.md

Transformers

build passing | license Apache-2.0 | website online | release v2.3.0

State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch

🤗 Transformers (formerly known as `pytorch-transformers` and `pytorch-pretrained-bert`) provides state-of-the-art general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, CTRL...) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over 32+ pretrained models in 100+ languages and deep interoperability between TensorFlow 2.0 and PyTorch.

Features

- As easy to use as pytorch-transformers
- As powerful and concise as Keras
- High performance on NLU and NLG tasks
- Low barrier to entry for educators and practitioners

State-of-the-art NLP for everyone

- Deep learning researchers
- Hands-on practitioners
- AI/ML/NLP teachers and educators

東北大から
MeCab Tokenize
の pre-trained モデルも提供

Hugging Face
@huggingface

おはようございます、日本の友達

Hello, Friends from Japan 🇯🇵!

Thanks to @NlpTohoku, we now have a state-of-the-art Japanese language model in Transformers, `bert-base-japanese`.

Can you guess what the model outputs in the masked LM task below?

```
import torch
from transformers.modeling_bert import BertForMaskedLM
from transformers.tokenization_bert_japanese import BertJapaneseTokenizer

tokenizer = BertJapaneseTokenizer.from_pretrained("bert-base-japanese-whole-word-masking")
model = BertForMaskedLM.from_pretrained("bert-base-japanese-whole-word-masking")
model.eval()

input_ids = tokenizer.encode(f"""
    山田さんが{tokenizer.mask_token}を見たのはこれが初めてでした。巨大だった。
""", return_tensors="pt")

# This was the first time for Mr. Yamada to see [MASK]. It was huge.

masked_index = torch.where(input_ids == tokenizer.mask_token_id)[1].tolist()[0]

result = model(input_ids)
result = result[0][:, masked_index].topk(5).indices.tolist()[0]
for r in result:
    output = input_ids[0].tolist()
    output[masked_index] = r
    print(tokenizer.decode(output))

# [CLS] 山田 さん が ゴジラ を 見 た の は これ が 初めて でし た 。 巨大 だっ た 。 [SEP]
# [CLS] 山田 さん が 映画 を 見 た の は これ が 初めて でし た 。 巨大 だっ た 。 [SEP]
# [CLS] 山田 さん が これ を 見 た の は これ が 初めて でし た 。 巨大 だっ た 。 [SEP]
# [CLS] 山田 さん が それ を 見 た の は これ が 初めて でし た 。 巨大 だっ た 。 [SEP]
# [CLS] 山田 さん が 富士山 を 見 た の は これ が 初めて でし た 。 巨大 だっ た 。 [SEP]
```

README.md

Pretrained Japanese BERT models

This is a repository of pretrained Japanese BERT models. The pretrained models are available along with the source code of pretraining.

Update (Dec. 15 2019): Our pretrained models are now included in Transformers by Hugging Face. You can use our models in the same way as other models in Transformers.

Features

- All the models are trained on Japanese Wikipedia.
- We trained models with different tokenization algorithms.
  - `mecab-ipadic-bpe-32k` : texts are first tokenized with MeCab morphological parser and then split into subwords by WordPiece. The vocabulary size is 32000.
  - `mecab-ipadic-char-4k` : texts are first tokenized with MeCab and then split into characters (information of MeCab tokenization is preserved). The vocabulary size is 4000.
- All the models are trained with the same configuration as the original BERT; 512 tokens per instance, 256 instances per batch, and 1M training steps.
- We also distribute models trained with **Whole Word Masking** enabled; all of the tokens corresponding to a word (tokenized by MeCab) are masked at once.
- Along with the models, we provide tokenizers, which are compatible with ones defined in Transformers by Hugging Face.

Pretrained models

- BERT-base models (12-layer, 768-hidden, 12-heads, 110M parameters)
  - `BERT-base_mecab-ipadic-bpe-32k.tar.xz` (2.1GB)
    - MeCab + WordPiece tokenization.
  - `BERT-base_mecab-ipadic-bpe-32k_whole-word-mask.tar.xz` (2.1GB)
    - MeCab + WordPiece tokenization. Whole Word Masking is enabled during training.
  - `BERT-base_mecab-ipadic-char-4k.tar.xz` (1.6GB)
    - Character tokenization.
  - `BERT-base_mecab-ipadic-char-4k_whole-word-mask.tar.xz` (1.6GB)
    - Character tokenization. Whole Word Masking is enabled during training (word boundaries are determined by MeCab).

1年以内に,BERT 改良モデルが続々登場

https://github.com/thunlp/PLMpapers

NEW

GPT-3 [Brown+ (OpenAI), 2020]

- 前身のGPT-1(1億), GPT-2(15億)と同様に自己回帰言語モデル(1,750億)
- タスクの説明もテキストとして入力し, マルチタスクを実現
- 少数のデモンストレーションに基づく転移学習が可能
  - **Few-shot**: 学習後はパラメータを更新せず, 推論時にタスク説明と少数 (10から100)のデモンストレーションを与える
  - **One-shot**: タスク説明と1つのデモンストレーションを与える
  - **Zero-shot**: タスク説明のみ与え全くデモンストレーションを与えない

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:      ← task description
cheese =>                         ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:      ← task description
sea otter => loutre de mer        ← example
cheese =>                         ← prompt
```

Few-shot

NEW

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:      ← task description
sea otter => loutre de mer        ← examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                         ← prompt
```

NEW

Vision Transformer (ViT) [Dosovitskiy+, 2021]

- Transformer は画像認識などの NLP以外でも成果を発揮

- 画像パッチを単語とみなす 632M の Transformer エンコーダ
- 画像は最初にパッチに分割した後, 線形変換で埋め込み
- 3億枚以上の画像分類で事前学習し, ImageNet 等で SOTA

https://github.com/google-research/vision_transformer

NEW

DALL・E [Radford+ (OpenAI), 2021]

- OpenAI が発表した文章に忠実な画像を生成するモデル

TEXT PROMPT

チュチュを着た赤ちゃん大根が犬を散歩させている

AI-GENERATED IMAGES

Edit prompt or view more images↓

TEXT PROMPT

アボカドの形をした肘掛け椅子

AI-GENERATED IMAGES

Edit prompt or view more images↓

文字を含む画像も生成できている

- 巨大な Transformer デコーダ による Text-to-image モデル
  - パラメタ数は最大12B (ViT の約20倍)
- 大量の画像と説明文ペアから学習, 生成画像のレベルが高い
- 画像は1024(32x32)のコード系列(8192種)として扱う

https://openai.com/blog/dall-e/

NEW

CLIP [Radford+ (OpenAI), 2021]

- 大規模な画像とテキストのペアで zero-shot の物体認識を実現

- 画像とテキストのマッチングを4億ペアから事前学習
  - DALL·E の生成画像のリランキングにも使われている
- 正しい画像・テキストペアを分類できるように Contrastive pre-training を行う

https://openai.com/blog/clip/

NEW

まとめ

- **深層学習の発展とともに自然言語処理も進化**
  - 応用タスクの学習データで End-to-end で学習可能
  - ブラックボックスのため, 出力の解釈が難しい等の課題もある
- **自然言語処理研究を超えて Transformer が席巻中**
  - BERT (Transformer) テキスト分類, 機械翻訳, 質問応答(機械読解), 要約, 文生成 など様々な自然言語処理の応用タスクの性能を向上
  - GPT-3 などで使われている Transformer が, 画像認識など自然言語処理以外の領域でも成果を発揮しつつある
- **トレンドは 視覚+言語の事前学習モデル** で,研究が盛ん

スケジュール

- 1日目: 6/25(金)
  - 説明 — テキストマイニングの手順
  - 説明 — データをよく知る (Excel)
- **2日目: 7/2(金)**
  - **説明 — テキストマイニングツールの使い方 (KHCoder)**
- 3日目: 7/9(金)
  - 説明 — データ分析の実践 (KHCoder)
  - 実習 — データ分析の実践 (KHCoder)
- 4日目: 7/16(金)
  - Text Mining Studio 利用体験
  - 実習 — データ分析の実践 (KHCoder)
- 体育の日: 7/23(金)
- 5日目: 7/30(金)
  - 発表 — データ分析の実践 (KHCoder)

※ お知らせ ※

3日目, 4日目後半, 5日目 は, Zoom のブレイクアウトルーム機能を使ったグループワークになります。

(復習) テキストマイニングの手順

- **データをよく知る**
  - データ件数や構成比を集計 → データを理解する
    - 旅行目的別の人気エリアは?
    - 同伴者別の人気エリアは?
    - 数値評価による人気エリアの差異は?
- **テーマを設定する**
  - 解決すべき課題を決める → 分析目的を明確にする
    - 数値評価が低い原因は?
    - 高評価の施設に学ぶ改善点は?
- **データ分析に取り組む**
  - これら課題を解決するために,テキスト分析を実施

(復習) 実習で使用するデータ

A. 楽天トラベル の**クチコミデータ**

- 収集期間は **2018-2019** および **2020-2021**(~5/12) の **2セット**
- エリアごと同数に **1,000件ずつ** ランダムサンプリング
- データ件数は **1万件** × 2セット

B. ハッシュタグ「#新型コロナ」で投稿された**ツイート**

- 収集期間は **2020-04-24 ~ 2021-05-05** の間で **1セット**
- Twitter APIで収集した **1%** サンプリングデータから **RTや重複を除去**
- データ件数は 32万件 からランダムサンプリングした **1万件**

(復習) A. クチコミデータ

- 楽天トラベルから収集した「お客様の声」のデータ
  - 宿泊日が **2018-2019**年 および **2020-2021**年(~5/12),下記10エリア

レジャー	5エリア	登別, 草津, 箱根, 道後, 湯布院	**1,000件** × 10エリア = 計10,000件
ビジネス	5エリア	札幌, 名古屋, 東京, 大阪, 福岡	

  - データ項目

施設情報	4項目	カテゴリ, エリア, 施設番号, 施設名
口コミ	1項目	コメント
ユーザー評価	7項目	総合, サービス, 立地, 部屋, 設備・アメニティ, 風呂, 食事
その他の分類	2項目	旅行の目的, 同伴者
宿泊日	1項目	宿泊年月
ユーザー情報	3項目	ユーザー, 年代, 性別

施設紹介 | プラン一覧 | フォトギャラリー(76) | 地図・アクセス | お客さまの声(886) | クーポン一覧 | プレゼント

鴨川シーワールドホテルのクチコミ・お客さまの声

ホテル・旅行のクチコミTOPへ

総合評価 4.12
アンケート件数：886件

評価内訳

5点	236件
4点	302件
3点	47件
2点	15件
1点	9件

項目別の評価

サービス	4.11
立地	4.61
部屋	3.53
設備・アメニティ	3.62
風呂	3.53
食事	4.10

総合 2
投稿者さんの 鴨川シーワールドホテル のクチコミ (感想・情報)

投稿者さん
2015年06月11日 17:03:57

良かったところ
・部屋からの景色（朝日最高でした）
・食事（品数も多く、朝夕とも良かったです）
・フロントの方の対応（お姉さんがとても頑張っていました）以上。
掃除が行き届いているとの口コミを多く見ましたが、そうは思いませんでした。
気にかかることは多々ありましたが、フロントのお姉さんが一生懸命で、その笑顔に救われた思いです。

評価	総合	2
	サービス	2
	立地	4
	部屋	4
	設備・アメニティ	2
	風呂	2
	食事	4
旅行の目的	レジャー	
同伴者	家族	
宿泊年月	2015年06月	

鴨川シーワールドホテル
2015年06月11日 19:32:50

この度は、ご利用頂きまして誠にありがとうございました。

客室内清掃の件、大変申し訳
重要改善として、早急に対応いたします。
今後は、この様な事の無いように、清掃・点検を
強化いたします。

フロントスタッフへのお言葉
誠にありがとうございます。
モチベーションアップに繋が
お客様からの声として、
スタッフと共有させて頂きます。

機会がございましたら、またご利用をお待ちしております。

テキストデータ

数値評価

(復習) データをよく知る ― まとめ

	データの特徴	テキスト分析時に注意すべき点
年代別・性別	・ 約60%が年代や性別を表明していない ・ 年代別では,目的によらず40～60代が多い ・ 全体的に男性の投稿者が多い (女性の倍程度) ・ レジャーに比べてビジネス方が男女差が大きい ・ レジャーの中でも男女差が大きいのは道後	・レビュー観点がある年代や性別に偏っている可能性 ・無回答(na)中が,ある年代や性別に偏っている可能性
目的別	・レジャーは家族が多い,ビジネスは一人が多い (出張は単独) ・レジャーの中でも,道後は男性の一人客が多い (道後はもはや仕事で行く場所)	・レビューの観点が性別によって偏っている可能性 ・レビューの観点がカテゴリと一致していない可能性 (道後→仕事)
数値評価 (総合)	・旅行目的によらず評価は高め ・レジャーがビジネスより評価が高め ・レジャーの中で高評価が多いのは湯布院,少ないのは登別 ・ビジネスの中で高評価が多いのは大阪と札幌, 少ないのは東京都と名古屋だが僅差	・好評価しか投稿しない→コメントが好評価に偏っている可能性 ・旅行目的によって投稿の動機が異なっている可能性
数値評価 (項目ごと)	・レジャーの評価は, 風呂や食事 > 設備や部屋 ・ビジネスの評価は, 立地 > その他 ・レジャーの中で湯布院は軒並み高評価 ・レジャーもビジネスも立地は高評価	・旅行目的によって評価の観点や重みが異なっている可能性

全体	・あくまでも楽天トラベルの特性であるので,旅行者の傾向として主張するためには別途裏付けが必要

(復習) 数値評価で違いを見るのは難しい

- ユーザーの 8割が 4～5 の評価, 1～2をつけない →本音が見えない

数値評価の平均 (エリア別)

行ラベル	平均 / サービス	平均 / 立地	平均 / 部屋	平均 / 設備･アメニ	平均 / 風呂	平均 /	平均 / 総合
A_レジャー	**4.24**	**4.28**	**4.12**	**4.05**	**4.32**	**4.24**	**4.29**
01_登別	4.13	4.24	3.97	3.96	4.38	4.10	4.18
02_草津	4.21	4.32	4.02	3.97	4.34	4.13	4.26
03_箱根	4.17	4.15	4.11	3.97	4.18	4.22	4.17
04_道後	4.18	4.38	4.14	4			4.30
05_湯布院	4.52	4.32	4.39	4			4.52
B_ビジネス	**4.09**	**4.38**	**4.20**	**4.03**	**3.91**		**4.28**
06_札幌	4.16	4.38	4.22	4.10	3.95	4.07	4.33
07_名古屋	4.10	4.31	4.16	3.98	3.89	3.96	4.25
08_東京	4.01	4.37				4.02	4.22
09_大阪	4.12	4.43				4.07	4.36
10_福岡	4.07	4.39				4.09	4.25

- 同じ点数でもテキストを見れば差異があるかも

- すべての項目に回答する → どこに注目しているかよくわからない

数値評価の平均 (レジャー,ビジネス別)

行ラベル	平均 / サービス	平均 / 立地	平均 / 部屋	平均 / 設備･アメニ	平均 / 風呂	平均 / 食事	平均 / 総合
A_レジャー	4.24	4.28	4.12	4.05	4.32	4.24	4.29
B_ビジネス	4.09	4.38	4.20	4.03	3.91	4.04	4.28

全学計算機システムのリモートデスクトップ

- 全学計算機システムのリモートデスクトップを使用します
  - 【Win】 https://www.u.tsukuba.ac.jp/remote/**#win2win**
  - 【Mac】 https://www.u.tsukuba.ac.jp/remote/**#mac2win**

上記のページにある説明に従って, 全学計算機システム(Windows)へログインができることを確認してください

Mac の場合:
左記のページにある説明に従って,事前にツール Microsoft Remote Desktop のインストールが必要です

KH Coder インストール時の注意:
全学の Windows の場合は,ログイン後の**デスクトップ上に「khcoder」というフォルダを作成**して,その中に解凍してください ※Cドライブへの保存は禁止されています

KH Coder のインストール

- ダウンロードとインストール **https://khcoder.net/dl3.html**

① ここをクリックすると遷移先のページからダウンロードが始まります

② ダウンロードしたファイルを実行（ダブルクリックし, 開いた画面上の「Unzip」ボタンをクリックします。

③ 保存先を「**Cドライブ以外**」(Cドライブへの保存は禁止されています)に変更します。例)「**Z:¥Desktop¥khcoder3**」

④ 指定した保存先フォルダにすべてのファイルが解凍されます。解凍された「**kh_coder.exe**」を実行すると KH Coder が起動します。

KH Coder —立命館の樋口先生が開発

- 社会調査データを分析するために開発されたフリーのテキストマイニングツール
  - 高機能,商用可能でフリー
  - Rを用いた多変量解析と可視化
  - 実装されている分析手法
    - 階層的クラスター分析
    - 多次元尺度構成法(MDS)
    - 対応分析
    - 共起ネットワーク
    - 自己組織化マップ
    - 文書のクラスター分析

論文検索サービスも提供 →
http://khcoder.net/bib.html?year=2018&auth=all&key=

※ 2021/6/22 現在 (→1646→2042→2695→昨年3741件→4554件)

KH Coder の情報

ホームページ http://khcoder.net/

参考書

チュートリアル
http://khcoder.net/tutorial.html

PDFファイルをダウンロードすることもできます。
※Windows版パッケージにはあらかじめ同梱してあります。

チュートリアルの実行に必要なデータファイルです。
※Windows版パッケージには同梱してありますので、別途ダウンロードする必要はありません。

参考 — KH Coder の分析手順

http://khcoder.net/diagram.html

文の出現パターンと単語の出現パターン

【行】ある文中に出現する単語の数を要素とする (文ベクトル)

【列】全文中に出現する単語の数を要素とする (単語ベクトル)

h5	bun	部屋	ホテル	風呂	温泉	お部屋	スタッ	立地	フロン	最高	浴場	お湯	露天風	感じ	夕食	バス	バイキ	家族	場所	トイレ	子供	ベット	コンビ	良い
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	6	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

距離で「似てる」を図る

- Jaccard 距離: KHCoder で標準的な距離尺度
  - 1 つ文書に含まれる**語が少ない**ケースや,各語が**一部の文書中**にしか含まれていないケースに向いている →スパースなデータ分析向き

<table>
<tr><th>Jaccard 距離 (2値)</th><th>ユークリッド距離</th><th>コサイン距離</th></tr>
<tr><td rowspan="2">• 1 つの文書の中に語が1 回出現した場合も 10 回出現した場合も単に「出現あり」(2値)と見なしてカウントした語と語の共起数を計算
• 語Aと語Bのどちらも出現していない文書(0-0対)が沢山あっても語Aと語Bが類似しているとは見なさない</td><td colspan="2">• 文書中における語の出現回数 (1,000 語あたりの出現回数に調整) を計算
• 1つひとつの文書が長く,多数の文書に含まれている語が多いデータ向き(各文書中での語の出現回数の大小が重要な場合)</td></tr>
<tr><td>• サイズの差までも見る場合向き</td><td>• 傾きが似ているかどうかだけを見る場合向き</td></tr>
<tr><td>
<table>
<tr><th></th><th>1</th><th>0</th></tr>
<tr><td>1</td><td>n_{11}</td><td>n_{10}</td></tr>
<tr><td>0</td><td>n_{01}</td><td>n_{00}</td></tr>
</table>
$${}_{J}s = \frac{n_{11}}{n_{11} + n_{10} + n_{01}}$$
</td><td>$${}_{E}d(\mathbf{x},\mathbf{y}) = \sqrt{\sum (x_i - y_i)^2}$$</td><td>$${}_{Cos}s(\mathbf{x},\mathbf{y}) = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2 \sum y_i^2}}$$</td></tr>
</table>

http://mjin.doshisha.ac.jp/R/68/68.html

KH Coder —スクリーンショット

階層的クラスター分析

抽出語の階層的クラスター分析を行い、デンドログラムを表示します。抽出語だけでなくコーディング結果（コード）についても、同じように分析を行えます。

New! デンドログラム

抽出語は出現数や品詞で選択

コードはチェックボックスで直接選択

対応分析

同じく抽出語またはコードを用いての、対応分析です。

同時布置図

New! バブルプロット

複数の外部変数を用いた多重対応分析

自己組織化マップ

抽出語またはコードを用いての、自己組織化マップです。

クラスター色分け

頻度のプロット

U-Matrix

多次元尺度構成法（MDS）

同じく抽出語またはコードを用いての、多次元尺度構成法です。

2次元の解

New! クラスタリングと色分け

3次元の解

共起ネットワーク

抽出語またはコードを用いて、出現パターンの似通ったものを線で結んだ図、すなわち共起関係を線（edge）で表したネットワークを描く機能です。

共起の程度が非常に強いものだけを線で結んだ図

やや弱い共起関係も描画に含め、自動的にグループ分け（色分け）

出現数が多い語ほど大きく、また共起の程度が強いほど太い線で描画

文書のクラスター分析

文書の分類を行うクラスター分析です。

クラスター分析の結果画面

併合水準のプロット。クラスター数5付近から併合水準が急上昇。10でも少し上がっているので、この場合クラスター数は11が良いか。

文書のデンドログラム。左の棒グラフは各文書の長さをあらわす。なお、文書数が500を超える場合、デンドログラムは表示不可。

KH Coder の主な分析手法

分析手法	解説
階層的クラスター分析	• 出現パターンの似た**単語**同士を**グルーピング(クラスタリング)**したもの • 出現パターンは,ある単語がどの文書に出現したかといった単語ベクトルで表現 • 類似度計算には Jaccard, ユークリッド, コサイン距離を用い,いわゆる Ward法, 群平均法, 最遠隣法で樹形図を作成
多次元尺度構成法 (MDS)	• 出現パターンの似た**単語**同士を**近くに置くよう図示**したもの • 出現パターンは,ある単語がどの文書に出現したかといった単語ベクトルで表現 • 類似度計算には Jaccard, ユークリッド, コサイン距離を用い, クラシカル, Kruskal, Sammon 法のいずれかで2次元にプロット
対応分析	• 出現パターンの似た**単語や外部変数**を**近くに置くよう図示**したもの • 単語と単語または外部変数が同時に出現した頻度をクロス集計し,それぞれの相関が最大になるような2変数で数値化し, 2軸上にプロット (PCAが元の情報をそのまま可視化するのに対し,対応分析は似ているものを近くに表示する) • 外部変数も同時にプロット可能
共起ネットワーク	• 同時に出現した**単語**同士を**ネットワークで結んで図示**したもの • 同時に出現したかといった共起の有無を集計し,ネットワークを作成 • 関係の強さ Jaccard 係数で評価, サブグラフは媒介性, クラスタリング精度(エッジ内の密度の高さ)を使って検出
自己組織化マップ	• 出現パターンの似た**単語**同士を**近くに集めて図示**したもの • ニューラルネットワークを利用して近い単語を集める方法で,距離にはユークリッド距離を使い,クラスタリングは Ward法
文書のクラスター分析	• 似た**文書**同士を**グルーピング(クラスタリング)**したもの • 各文書は,文書中に出現する単語の有無でベクトル化した文書ベクトルで表現 • 類似度計算には Jaccard, ユークリッド, コサイン距離を使い, いわゆる Ward法, 群平均法, 最遠隣法で階層クラスタを作成

データの取得方法

- https://github.com/haradatm/lecture/tree/master/**gssm-202107**

ダウンロード方法

ダウンロードボタンをクリック

使い方 ― プロジェクトの作成

注1: 次回 KH Coderを起動した時は「新規」ではなく「開く」を選択します
注2: ②のファイル選択後,ここに「テキスト」等の選択項目が表示されるまで数分がかかります

①メニューから「プロジェクト」「新規」を選択 (注1)

②「参照」をクリックして「rakuten-1000-2020-2021.xlsx」を開く

③「コメント」を選択 (注2)

④説明「楽天トラベル」等を入力

⑤「OK」をクリック

使い方 — 前処理 (形態素解析)

注1: EXCELファイルを読み込んで分析する場合,
あらかじめ「---cell---」が入力されています
注2: メニューから「前処理」「複合語の検出」を選ぶと,複合語候補の一覧を出力できます

①メニューから「前処理」「語の取捨選択」を選ぶ

②複合語など強制的に抽出したい語あればここに登録する (例:**湯畑**)

③削除語(ストープワード)など抽出したくない語あればここに登録する

④メニューから「前処理」「前処理の実行」を選ぶ

使い方 — 頻出語を確認する

	A	B	C	D	E	F	G	H
1	抽出語	出現回数		抽出語	出現回数		抽出語	出現回数
2	部屋	4878		前	714		非常	420
3	思う	4082		丁寧	677		ご飯	416
4	良い	4066		過ごす	670		悪い	410
5	利用	3746		素晴らしい	658		新しい	405
6	宿泊	2816		近く	649		高い	402
7	ホテル	2814		使う	612		助かる	402
8	風呂	2690		対策	608		狭い	389
9	食事	2581		過ごせる	607		種類	387
10	満足	2080		月	607		お湯	385
11	朝食	2071		家族	598		清掃	385
12	美味しい	2046		人	597		素敵	382
13	お部屋	1766		子供	570		楽しめる	374
14	温泉	1721		初めて	550		設備	374
15	スタッフ	1649		駐車	549		置く	364
16	対応	1623		バス	546		従業	363
17	行く	1408		入れる	536		気持ち	357
18	広い	1302		嬉しい	528		湯畑	355
19	大変	1253		旅行	519		掃除	354

使い方 — 共起ネットワークの作成

①メニューから「ツール」「抽出語」「共起ネットワーク」を選ぶ

② 「集計単位」として「文」を選んで 「OK」をクリック

③品詞を選択する,例えば
名詞, 名詞B, 名詞C, 未知語,
形容詞, 形容詞B, 形容動詞, タグ

④ 「調整」をクリックして「強い共起関係ほど…」をチェック

⑤ 「カラー」の箇所を「サブグラフ検出(modularity)」に変更

KH Coder の品詞体系

表 A.1　KH Coder の品詞体系

KH Coder 内の品詞名	茶筌の出力における品詞名
名詞	名詞–一般（漢字を含む 2 文字以上の語）
名詞 B	名詞–一般（平仮名のみの語）
名詞 C	名詞–一般（漢字 1 文字の語）
サ変名詞	名詞–サ変接続
形容動詞	名詞–形容動詞語幹
固有名詞	名詞–固有名詞–一般
組織名	名詞–固有名詞–組織
人名	名詞–固有名詞–人名
地名	名詞–固有名詞–地域
ナイ形容	名詞–ナイ形容詞語幹
副詞可能	名詞–副詞可能
未知語	未知語
感動詞	感動詞またはフィラー
タグ	タグ
動詞	動詞–自立（漢字を含む語）
動詞 B	動詞–自立（平仮名のみの語）
形容詞	形容詞（漢字を含む語）
形容詞 B	形容詞（平仮名のみの語）
副詞	副詞（漢字を含む語）
副詞 B	副詞（平仮名のみの語）
否定助動詞	助動詞「ない」「まい」「ぬ」「ん」
形容詞（非自立）	形容詞－非自立（「がたい」「つらい」「にくい」等）
その他	上記以外のもの

「KH Coder 3 リファレンス・マニュアル」
P.14 より

注: どの品詞を選択すべきかは,分析対象のデータや分析目的により異なります. 分析結果を確認しながら,適宜,適切な品詞選択を検討することが重要です

使い方 — 語句の前後文脈を表示する

①メニューから「ツール」「抽出語」「KWICコンコーダンス」を選ぶ

②検索したい語 (例:「風呂」) を入力して Enter キー

ダブルクリックで,さらに広い範囲の文脈を表示

③「集計」をクリックするとコロケーション統計(右)を開く

「右1」は右側の1つ目(=直後)に出現していた回数

「広い」は「風呂」の2語後に 84 回出現

④表示する語の品詞を選択 (例: 形容詞,形容詞B,形容動詞)

⑤「右合計」でソート

注: 共起ネットワーク上で「風呂」をクリックすると①②と同じ操作となります (V3以降)

使い方 ― 外部変数(エリア)を利用する

①メニューから「ツール」「外部変数と見出し」を開く

③「文」を選択

④「特徴語」「一覧(Excel形式)」を選択

	A	B	C	D	E	F	G	H	I	J	K
1											
2	01_登別			02_草津			03_箱根			04_道後	
3	食事	.068		湯畑	.066		食事	.071		部屋	.053
4	風呂	.058		風呂	.064		思う	.063		利用	.051
5	思う	.056		食事	.061		良い	.058		温泉	.048
6	宿泊	.049		良い	.058		風呂	.052		宿泊	.045
7	温泉	.045		温泉	.058		美味しい	.049		朝食	.042
8	美味しい	.043		草津	.052		露天風呂	.043		ホテル	.041
9	満足	.042		美味しい	.043		満足	.041		風呂	.037
10	残念	.034		満足	.042		お部屋	.039		美味しい	.035
11	行く	.034		宿	.038		温泉	.039		道後	.034
12	料理	.034		行く	.037		宿	.037		立地	.028
13	05_湯布院			06_札幌			07_名古屋			08_東京	
14	食事	.069		利用	.059		ホテル	.058		利用	.060
15	宿	.065		朝食	.059		利用	.057		部屋	.056
16	美味しい	.062		ホテル	.055		部屋	.056		ホテル	.054
17	良い	.057		部屋	.054		名古屋	.053		駅	.033
18	風呂	.056		思う	.051		朝食	.050		フロント	.033
19	温泉	.046		札幌	.050		良い	.045		朝食	.032
20	料理	.045		良い	.048		綺麗	.031		便利	.031
21	露天風呂	.044		宿泊	.045		対応	.031		立地	.031
22	満足	.044		立地	.033		快適	.029		快適	.029
23	お部屋	.041		対応	.031		駅	.029		広い	.027
24	09_大阪			10_福岡							
25	ホテル	.064		ホテル							
26	利用	.059		利用							
27	部屋	.056		部屋							
28	思う	.047		思う							
29	朝食	.042		朝食							

各エリアの特徴語を10件ずつ一覧 (数値は Jaccard係数)

使い方 — 対応分析による探索1

①メニューから「ツール」「抽出語」「対応分析」を選ぶ

②「文」になっていることを確認

③「抽出語×外部変数」「カテゴリー」「エリア」を選択

④「OK」をクリック

原点(0.0) から見て離れている語ほど特徴的

原点(0,0)

原点(0,0) 付近にあまり特徴のない語が集まっている

⑤「調整」をクリックして「バブルプロット」をチェック

使い方 ― コーディングルール

①メニューから「ツール」「文書」「文書検索」を選ぶ

ダブルクリックで,さらに広い範囲の文脈を表示可能

※ コーディングルール: 語ではなくコンセプトを数えるための方法

coding-rule.txt の中身

*ポジ
良い or 美味しい or 広い or 多い or 素晴らしい or 嬉しい or 気持ちよい or 楽しい or 近い or 大きい or 気持ち良い or 温かい or 早い or 優しい or 新しい or 暖かい or 快い or 明るい or 美しい or 可愛い

*ネガ
古い or 無い or 高い or 悪い or 小さい or 狭い or 少ない or 寒い or 遅い or 熱い or 欲しい or 暑い or 冷たい or 遠い or 臭い or 暗い

*風呂1-2
<>風呂-->1 | <>風呂-->2

*風呂4-5
<>風呂-->4 | <>風呂-->5

外部変数

使い方 — 対応分析による探索2

①メニューから「ツール」「コーディング」「対応分析」を選ぶ

②「文」になっていることを確認

③見たい「コード」を選択

④「抽出語×外部変数」「カテゴリー」「エリア」を選択

⑤「OK」をクリック

⑥「調整」をクリックして「バブルプロット」をチェック

使い方 — クロス集計

①メニューから「ツール」「コーディング」「クロス集計」を選ぶ

②「参照」をクリックして「coding-rule.txt」を開く

④「集計」をクリック

③「エリア」を選択

⑤「すべて」をクリック

課題 — 数値評価と口コミの傾向比較

- コーディングルール「coding-rule.txt」中の「風呂1-2」「風呂4-5」を参考に「総合1-2」「総合4-5」のルールを定義したコーディングルールを作成してください

  サンプル:

  https://github.com/haradatm/lecture/blob/master/gssm-202107/03-samples/**coding-rule_new.txt.zip**

- 前ページで紹介したクロス集計を行い,作成したプロットを **PDF** ファイルで提出してください

- 形式: PDF，提出先: manaba，期限: 次回 7/9 21:00

操作例 — 数値評価と口コミの傾向比較

①メニューから「ツール」「コーディング」「クロス集計」を選ぶ

② 「参照」をクリックして「coding-rule_new.txt」を開く

④ 「集計」をクリック

③ 「エリア」を選択

⑤ 「すべて」をクリック

実習 + Q&A

グループ分け

グループ1	202040166
	202040169
	202040176
	202140054
	202140072
グループ2	202040153
	202140053
	202140058
	202140077
	202140082
グループ3	201921622
	202020706
	202040171
	202140065
	202140066

グループ4	202121035
	202140057
	202140064
	202140070
	202140078
グループ5	202130165
	202140060
	202140062
	202140067
	202140081
グループ6	201945008
	202040081
	202040173
	202140063
	202140071

グループ7	202040174
	202140051
	202140068
	202140080
	202140084
グループ8	202140056
	202140059
	202140061
	202140073
	202140074

参考書

(KH Coder)

[1] 樋口耕一. 社会調査のための計量テキスト分析—内容分析の継承と発展を目指して【第2版】 KH Coder オフィシャルブック. ナカニシヤ出版, 2020.

[2] 樋口耕一. テキスト型データの計量的分析 —2つのアプローチの峻別と統合—. 理論と方法, 数理社会学会, 2004, 19(1): 101-115.

[3] 牛澤賢二. やってみよう テキストマイニング —自由回答アンケートの分析に挑戦!. 朝倉書店, 2019

(Windows環境によるデータ収集方法の参考に)

[4] テキストマイニングソフトを利用した新未来洞察手法の研究. 第10分科会, (財)市場創造研究会. http://www.shijo-sozo.org/news/第10分科会_1.pdf

参考書

(R を使った参考書)

[5] 金明哲. "テキストデータの統計科学入門." 岩波書店, 2009.

[6] 石田基広. "RMeCab によるテキスト解析. R によるテキストマイニング入門." 森北出版, 2008, 51-82.

(他のツールを使った参考書)

[7] 那須川哲哉. "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." 東京電機大学出版局, 2006.

[8] 上田隆穂, 黒岩祥太, 戸谷圭子. "テキストマイニングによるマーケティング調査." 講談社, 2005.

(統計解析を中心とした参考書)

[9] 前田忠彦; 山崎誠. 言語研究のための統計入門. くろしお出版株式会社, 東京, 2013.