

単語ベクトルの幾何と その周辺

統計数理研究所/国立国語研究所

持橋大地

daichi@ism.ac.jp

JAIST認知科学サマースクール2026

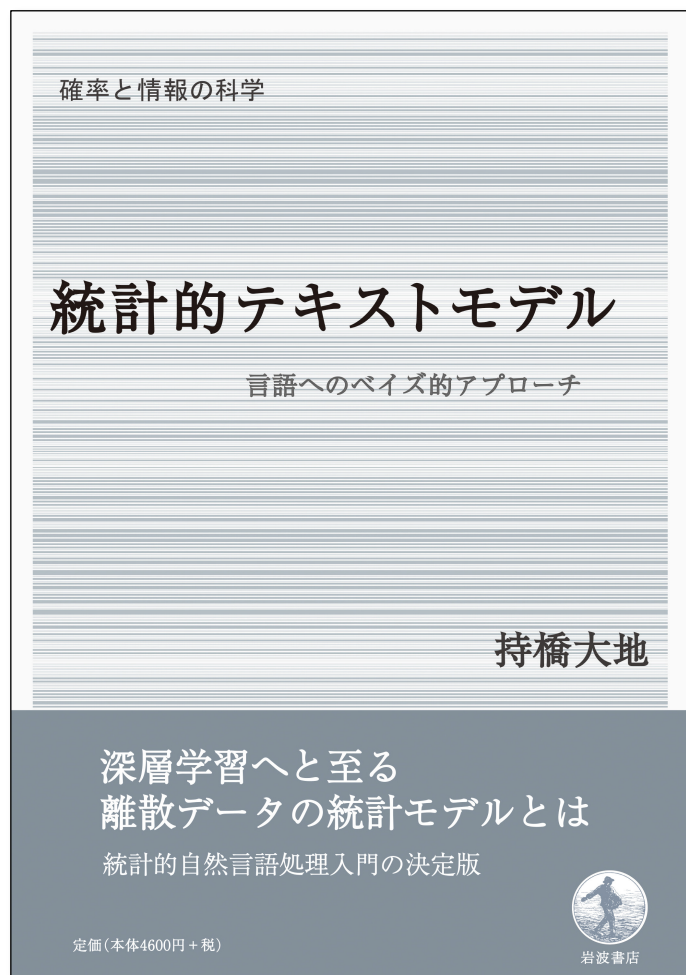
2026-9-15 (火)

自己紹介

- 統計数理研究所
統計基盤数理研究系・教授
- 国立国語研究所 次世代言語
科学研究センター教授 (兼務)
 国立国語研究所
次世代言語科学研究センター
- 東大文科三類→基礎科学科
→NAIST情報科学研究科で博士
- 専門：統計的自然言語処理、機械学習
最近はデータサイエンスが多い



『統計的テキストモデル』 岩波書店



- 2025/7/1発売
- 「ブラックボックスに頼らない」テキスト分析のための教科書
- まだAmazonのレビューが少ない (9件)
→可能な限り易しく書いているはず (?!)

3章 「単語の統計モデル」 より

- 日本語版text8からNPMIを用いて統計的に認識されたフレーズとその頻度 (最大 $2^4=16$ 単語まで結合)

頻度	フレーズ	
107101	し_た	:
70074	で_ある	45 全国_高等_学校_野球_選手権_大会
48360	さ_れ_た	37 お_ジャ_魔女_どれ_み
31219	て_いる	32 福島_第_一_原子力_発電_所_事故
28473	し_て_いる	27 ほっ_か_ほっ_か_亭
23371	さ_れ	26 福島_第_一_原子力_発電_所
:		23 連合_国軍_最高_司令_官_総_司令_部
1122	第_二_次_世界_大戦	19 学研_奈良_登美_ヶ_丘_駅
1094	と_なっ_て_い_た	16 国際_連合_安全_保障_理事_会
1092	最終_的	10 ソビエト_社会_主義_共和_国_連邦
1088	基本_的	10 自転車_競技_選手権_大会
:		10 工学部_機械_工学科
100	について_述べる	10 合衆国_最高_裁判所
100	キリスト_教徒	10 衆議院_予算_委員_会
100	陸上_競技_選手	10 大学院_工学_研究_科
100	ゆう_ちょ_銀行	

『確率的機械学習』 朝倉書店

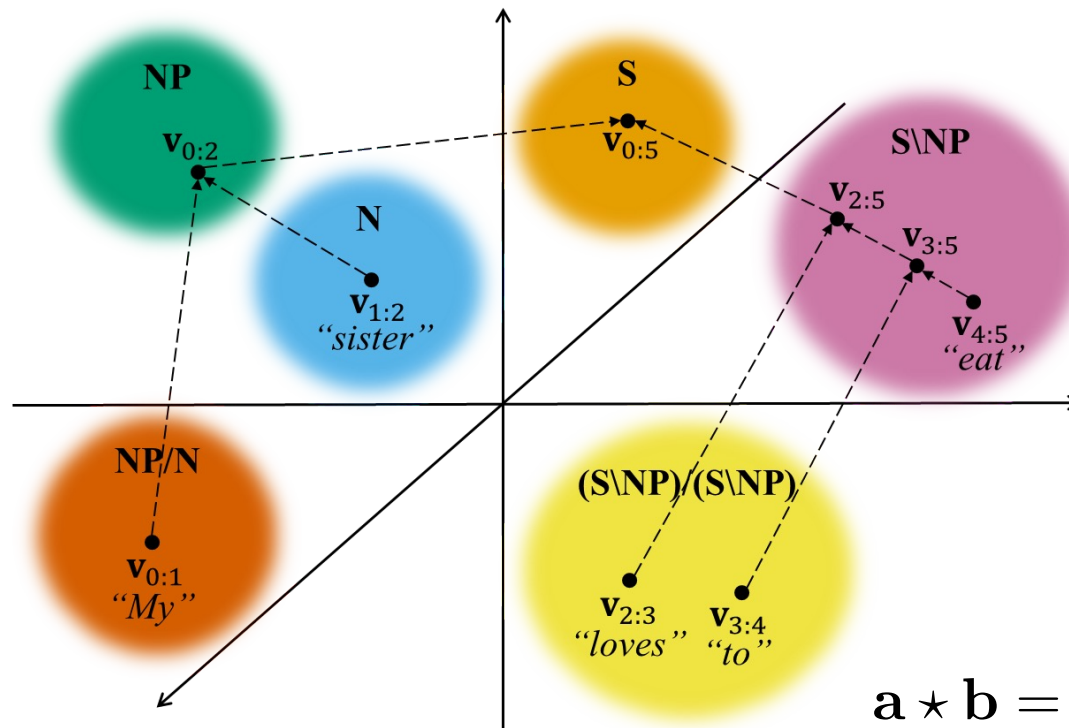
- 鈴木大慈さん(東大)と2人で監訳 (2025/10/25発売)
 - Kevin Murphy, “Probabilistic Machine Learning”翻訳



“次世代の
PRML”

Holographic CCG Parsing (ACL 2023)

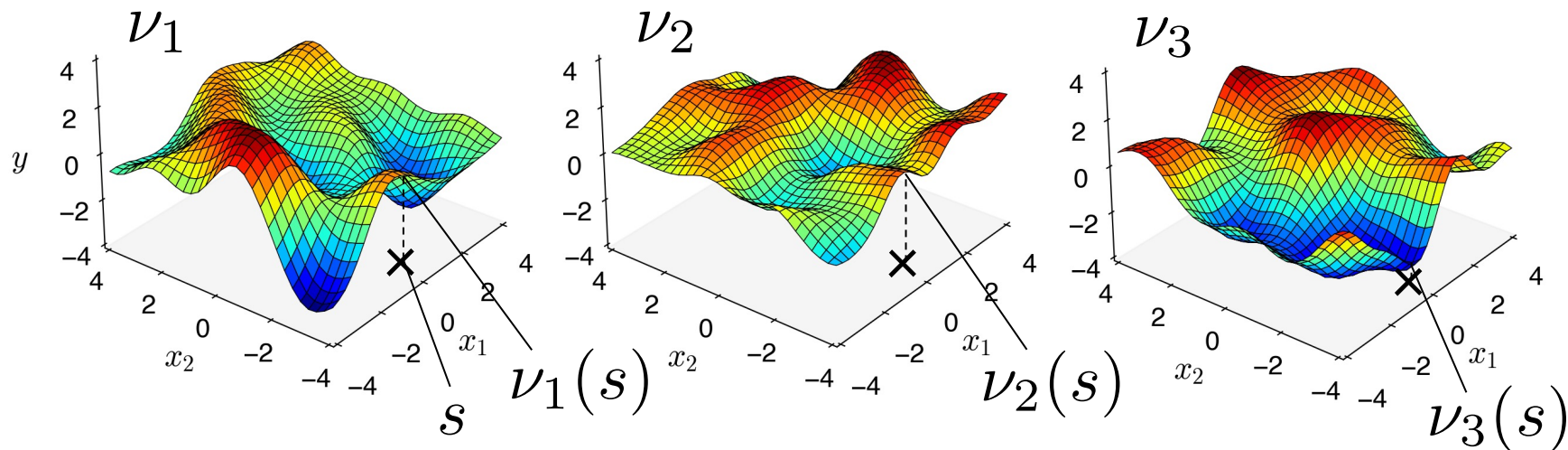
- 組み合わせ範疇文法(CCG)の構文解析をベクトル空間で行う



$$a \star b = \mathcal{F}^{-1}(\overline{\mathcal{F}(a)} \odot \mathcal{F}(b))$$

- 単語ベクトル自体を最適化し、合成して最終的に文Sのベクトルが得られるようにする

Spatial Indian Buffet process (投稿中)

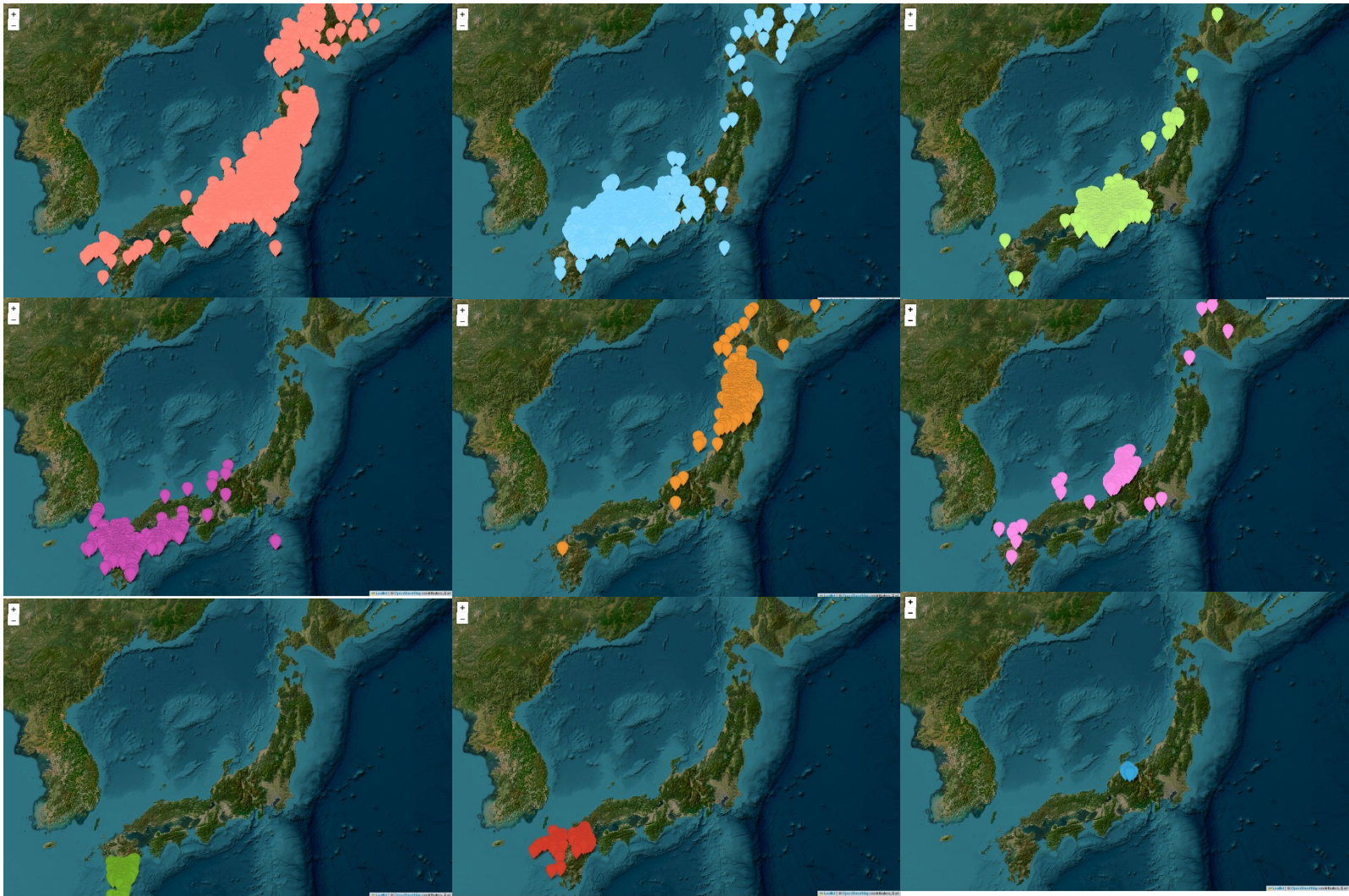


$$p(z_{ik}(s_i)=1) = \prod_{j=1}^k \sigma(\nu_j(s_i)), \quad \sigma(x) = \frac{1}{1+e^{-x}},$$

$$(\nu_j(s_1), \dots, \nu_j(s_N)) \sim \text{GP}(s_1, \dots, s_N)$$

- Now IBP becomes spatially dependent through a stick-breaking representation !

Latent factors from Linguistic Atlas of Japan



前半

大規模言語モデルと 単語ベクトルの幾何

自然言語処理の簡単な歴史

- ～2000年: 論理に基づく言語処理
 - 2000年～2010年頃: 統計的機械学習の時代
 - 2013年: Word2Vec登場、深層学習の始まり
 - 2017年: Transformer登場、高精度言語モデル
 - 2020年代: Transformerに基づく大規模言語モデル (LLM) の時代
- それぞれの時代に、過去を捨ててよいのか?
→ そうではない! (私見)

大規模言語モデルとは

- 大規模言語モデル=(おもにTransformerに基づく)
言語モデルを超大規模に学習したもの
- 言語モデルとは： 実は持橋の専門
単語列 $\mathbf{w} = w_1 w_2 \cdots w_T$ について、その確率 $p(\mathbf{w})$ を与えるモデルのこと

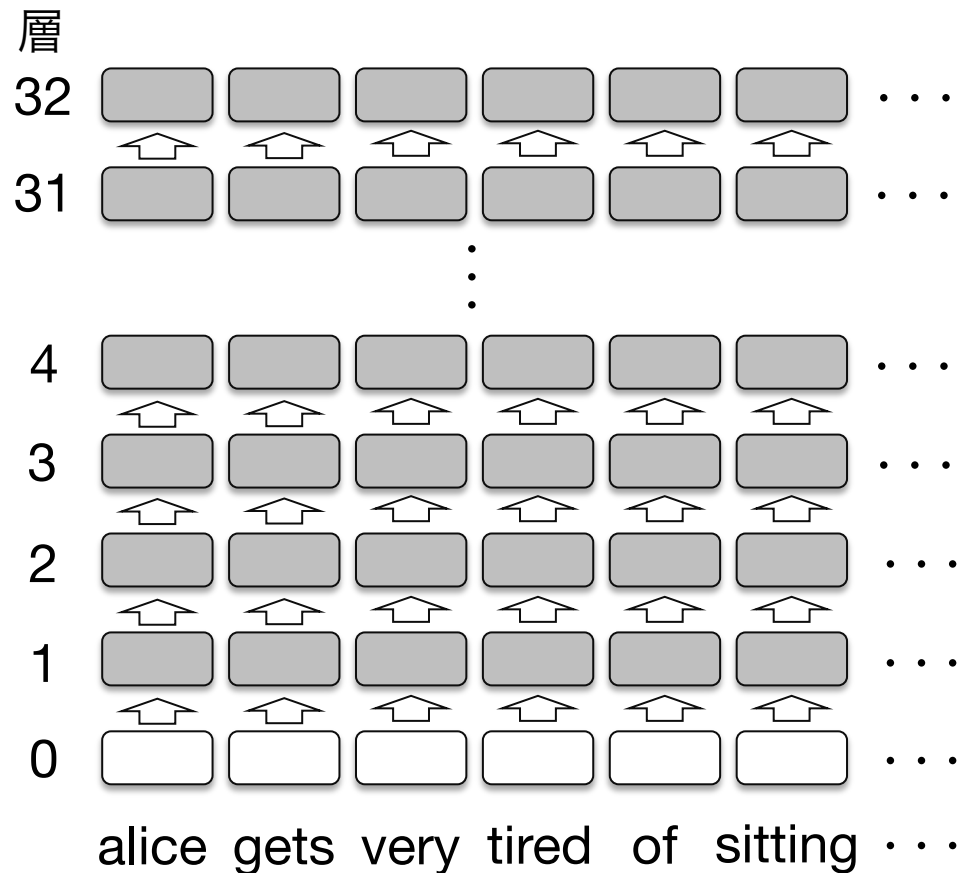
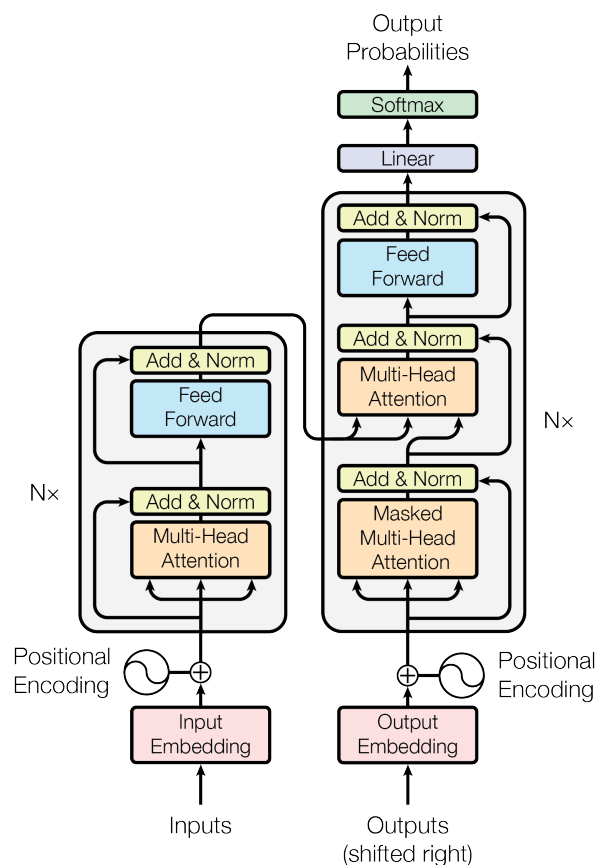
- 確率の連鎖率から、
$$p(\mathbf{w}) = \prod_{t=1}^T p(w_t | w_1 \cdots w_{t-1})$$

と分解できる→LLMはこの確率を実際に最大化

- 昔は、上の確率はTransformerではなく n グラムなどでモデル化されていた (持橋2007, ∞ -gram LM)
- 実際には単語ではなく、自動学習されるトークン列

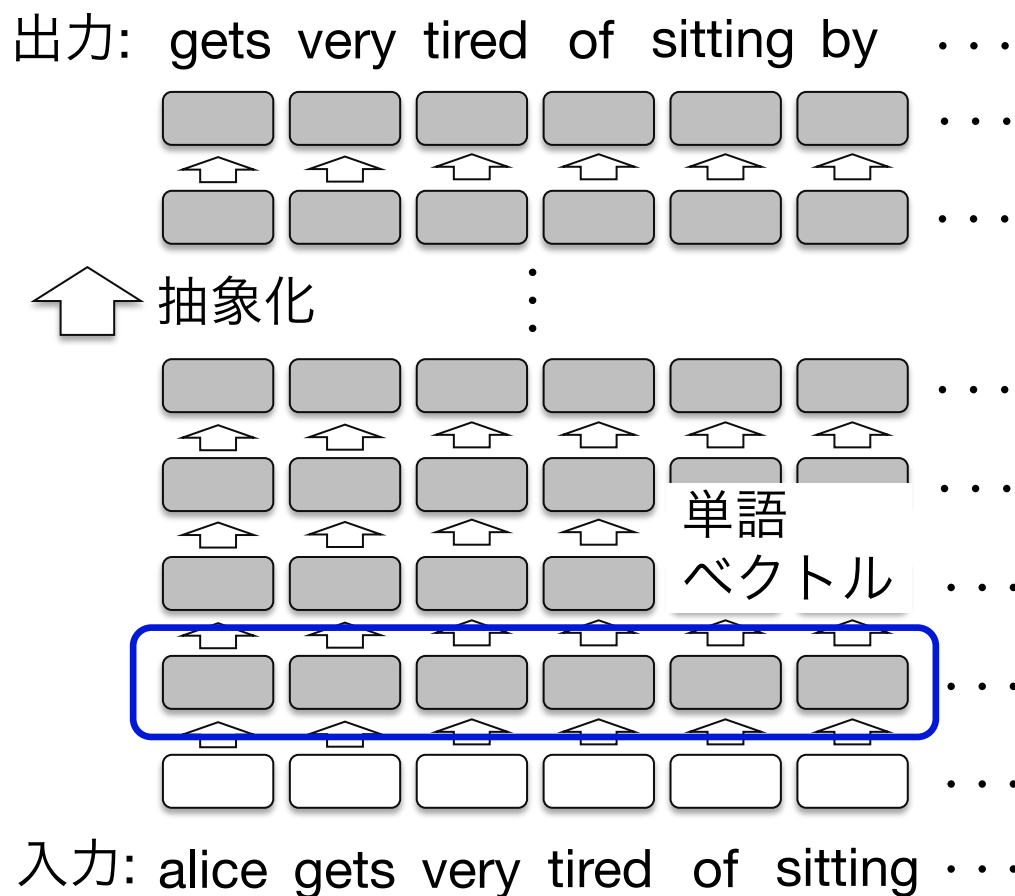
大規模言語モデル(LLM)とTransformer

- LLMのベースとなるTransformerは、一般的には左の構造の繰り返し→内部は右のようになっている



大規模言語モデルとTransformer (2)

- LLMの1層目およびその上層は, 実質的に単語ベクトル!



Output

31	文	:	_"	花
29	文	:	_"	花
27	文	:	_"	花
25	文	:	_flower	花
23	文	:	_flowe...	_flowe...
21	文	:	_"	_flowe...
19	文	:	_flowe...	_flowe...
17	eval	:	_"	<0xE5>
15	ji	:	_"	ψ
13	i	_vac	ols	_bore
11	eda	eda	_Als	abei
9	eda	ná	_Als	_hel
7	iser	arie	◀	arias
5	npa	orr	◀	arias
3	心	ures	_Bedeut	arda
1	_beskre	化	Portal	_Kontr...

中 文 :

Wender+(2024) “Do Llamas work in English?”
より引用

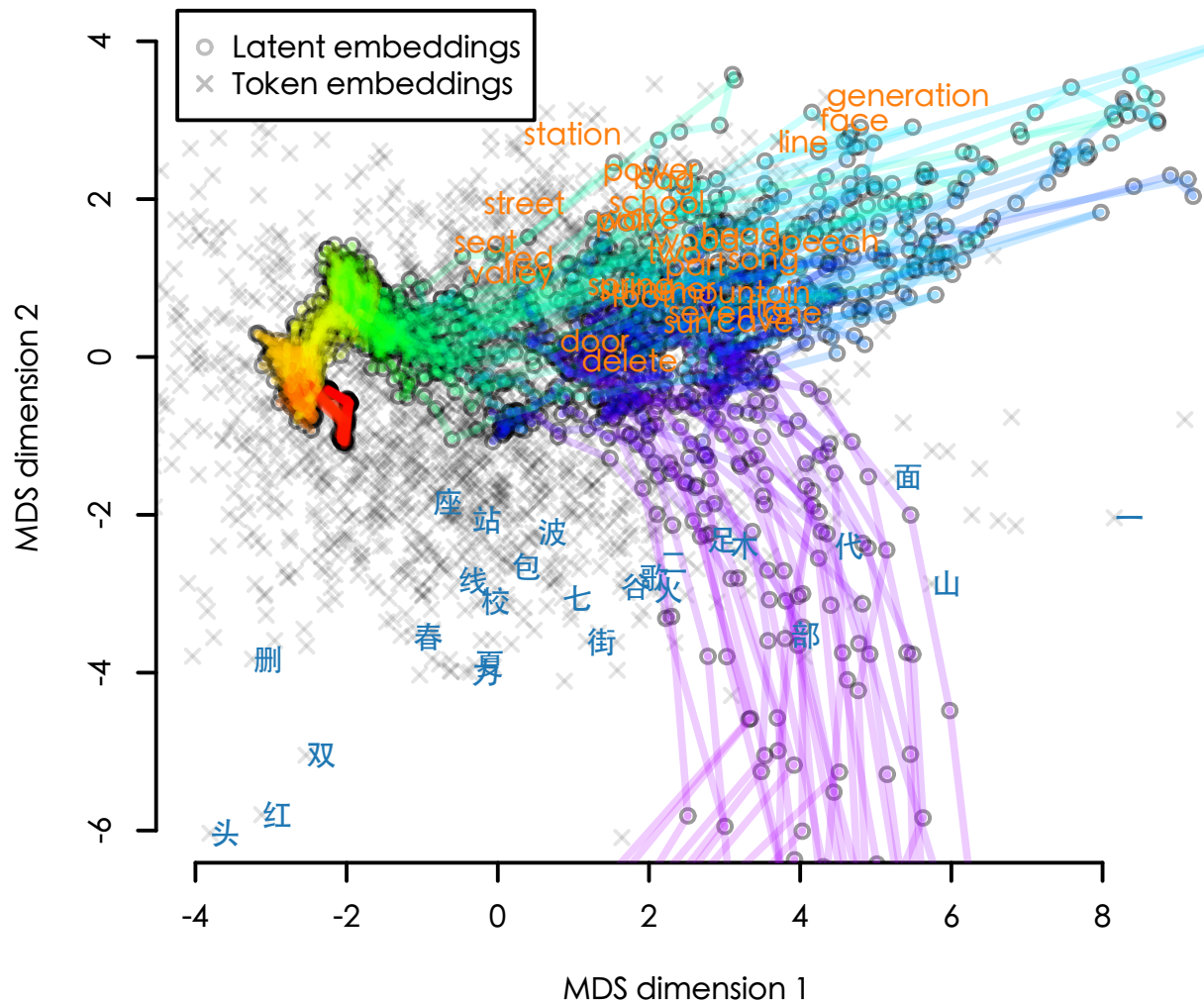
Transformerにおける潜在層の様子 [1]

Output	文	:	—"	花
31	文	:	—"	花
29	文	:	—"	花
27	文	:	__flower	花
25	文	:	__flowe...	__flowe...
23	文	:	—"	__flowe...
21	文	:	__flowe...	__flowe...
19	文	:	—"	__flowe...
17	eval	:	—"	<0xE5>
15	ji	:	—"	ψ
13	ī	__vac	ols	__bore
11	eda	eda	__Als	abei
9	eda	ná	__Als	__hel
7	iser	arie	◀	arias
5	npa	orr	◀	arias
3	心	ures	__Bedeut	arda
1	__beskre	化	Portail	__Kontr...
	中	文	:	—"

- 入力文
「中文：_」における各トークンの潜在層を、出力単語ベクトルで表現したもの (LogitLens)
- 32層の途中で、急に英語→中国語の順に回答が得られている

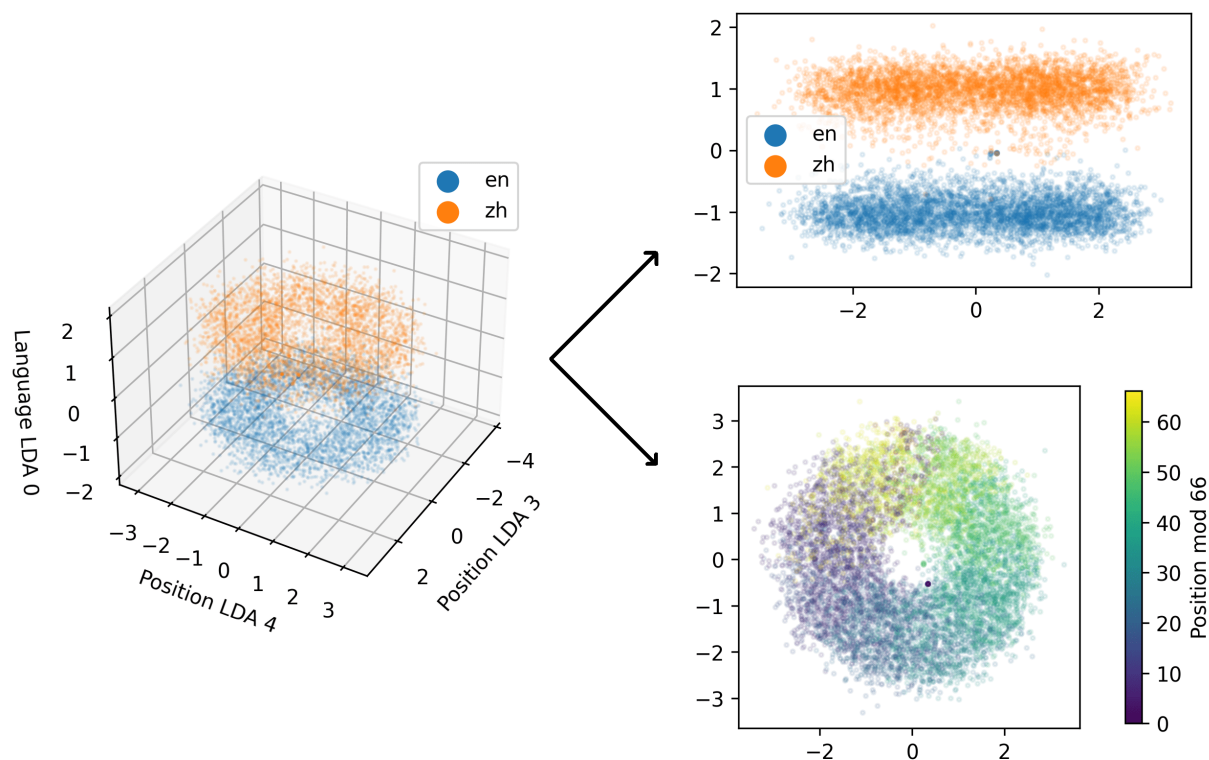


Transformerの潜在層と言語 [1]



- 潜在層のレイヤーが出力に近づくにつれ、いったん英語を介してから中国語の単語の確率が高まっている

Transformerと個別言語 [2]

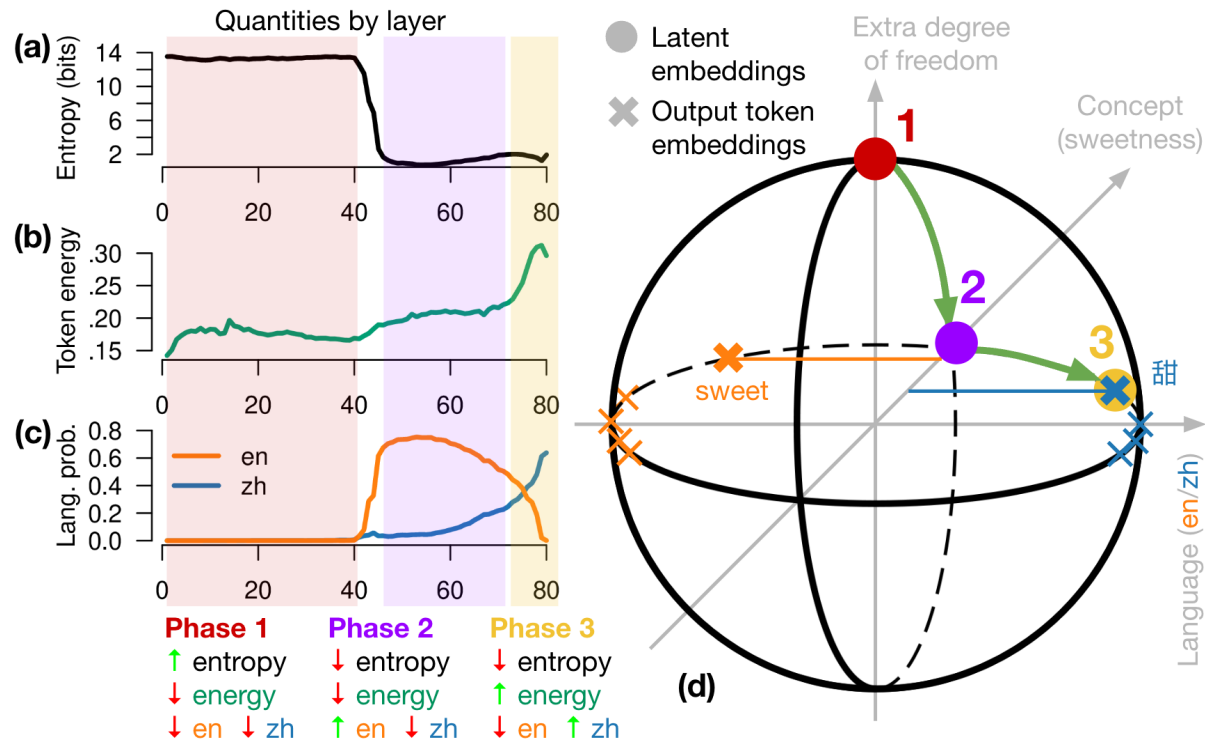


6層目の
中間層を
各言語のラベル
を使って
線形判別分析

- Transformerの潜在層にFisherの線形判別分析を行うと、英語と中国語に特化した部分空間が抽出できる
=language subspace

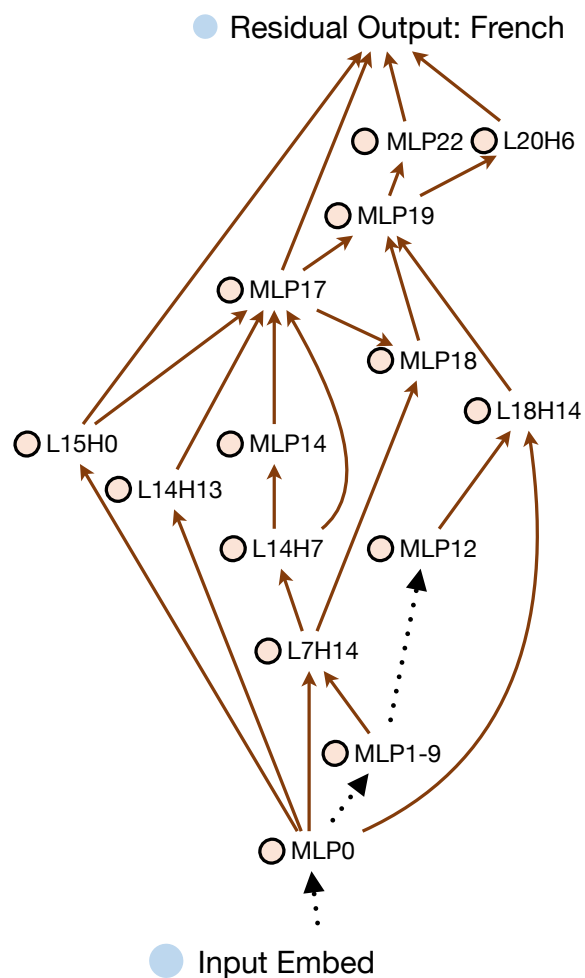


潜在層の計算のモデル [2]



- レイヤー正規化により各潜在層は超球面上にあるので、北極-南極を結ぶ大円上の「考える」フェーズと赤道上でそれを言語にマップするフェーズに分かれると考えられる

Transformerの「知識回路」 [3]

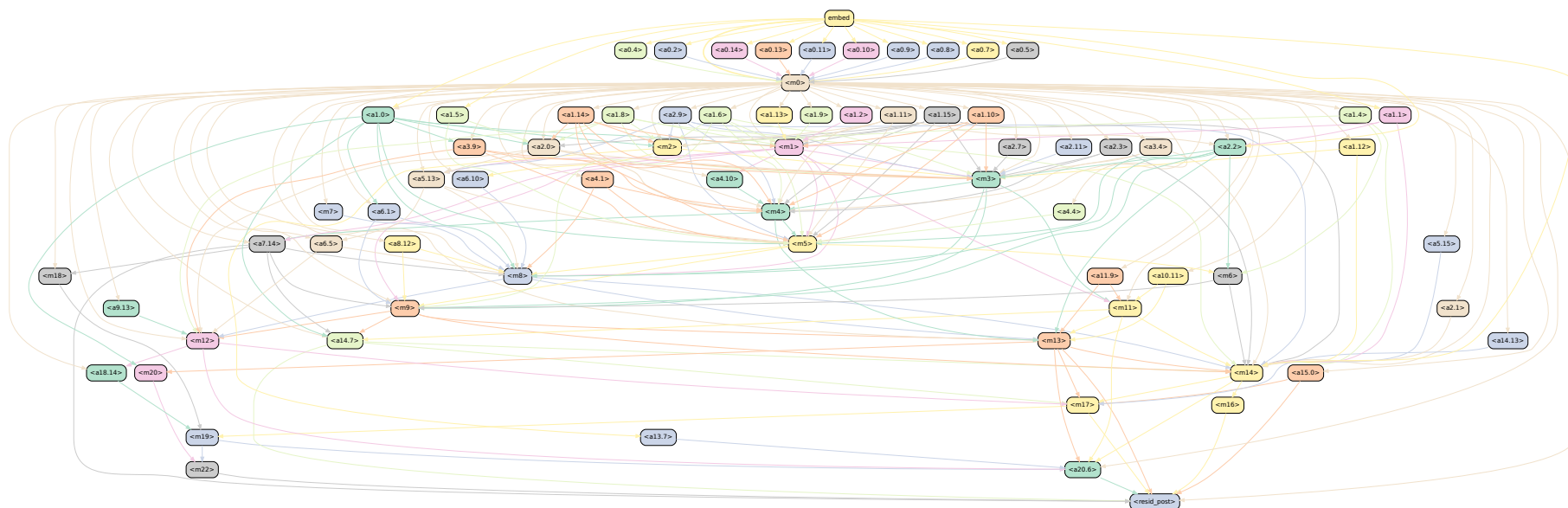


- GPT-2 Mediumで、“The official language of France is → French” を出力した回路
- Ablationスコア

$$S(e_i) = \log p(o|s, r) - \log p_{-e_i}(o|s, r)$$

の高い接続を残し、簡単化して表示したもの

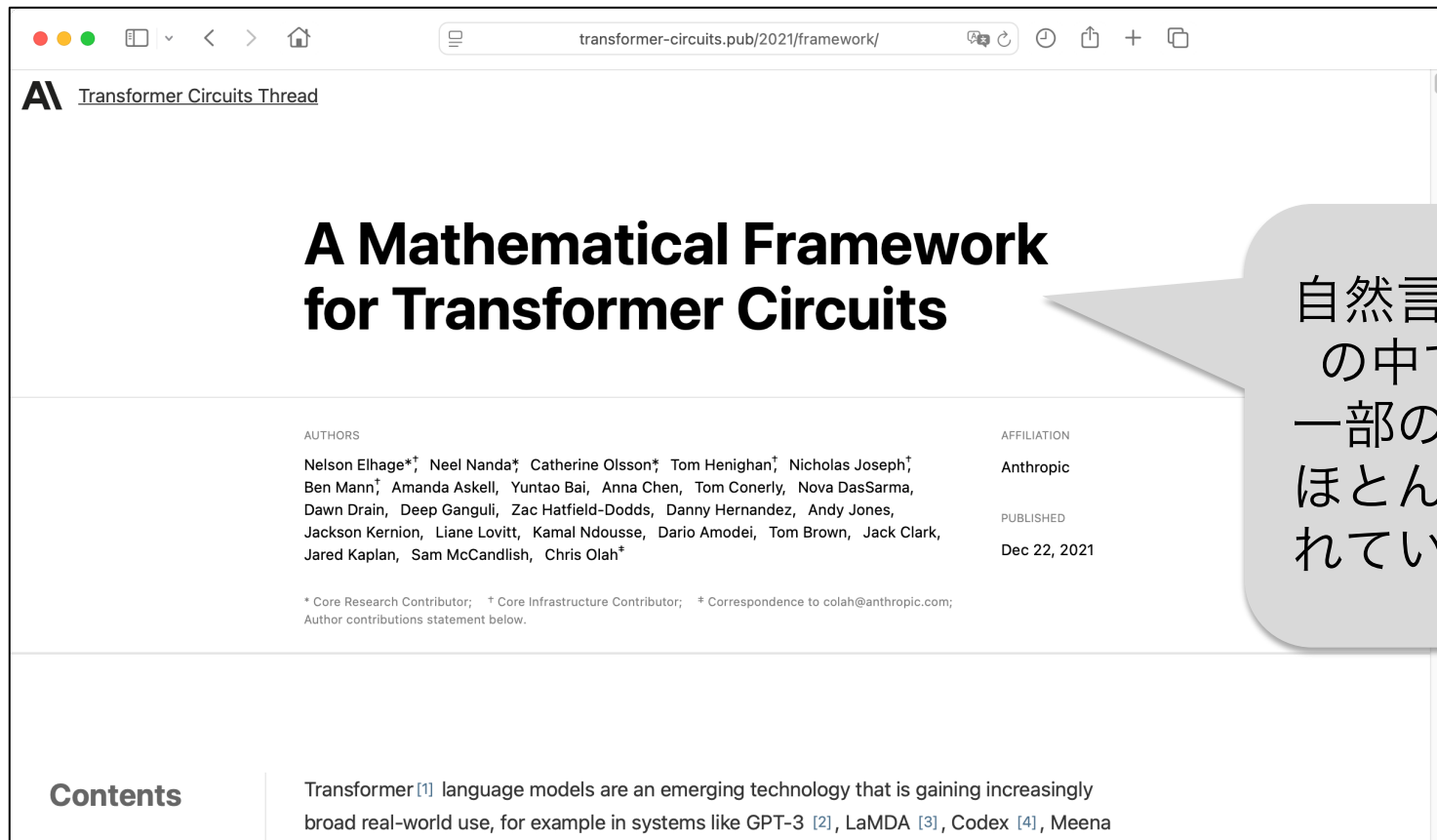
実際の知識回路 (GPT-2 Medium)



- 明らかに、回路としての理論的な扱いが必要
 - だが、自然言語処理分野にはそうした技術を持つ人はほとんどいない

分析の数学的フレームワーク [4]

- <https://transformer-circuits.pub/2021/framework/>



自然言語処理
の中では、
一部の人以上
ほとんど知ら
れていない！

[4] “A Mathematical Framework for Transformer Circuits”, Elhage+, Transformer Circuits Thread, 2021.



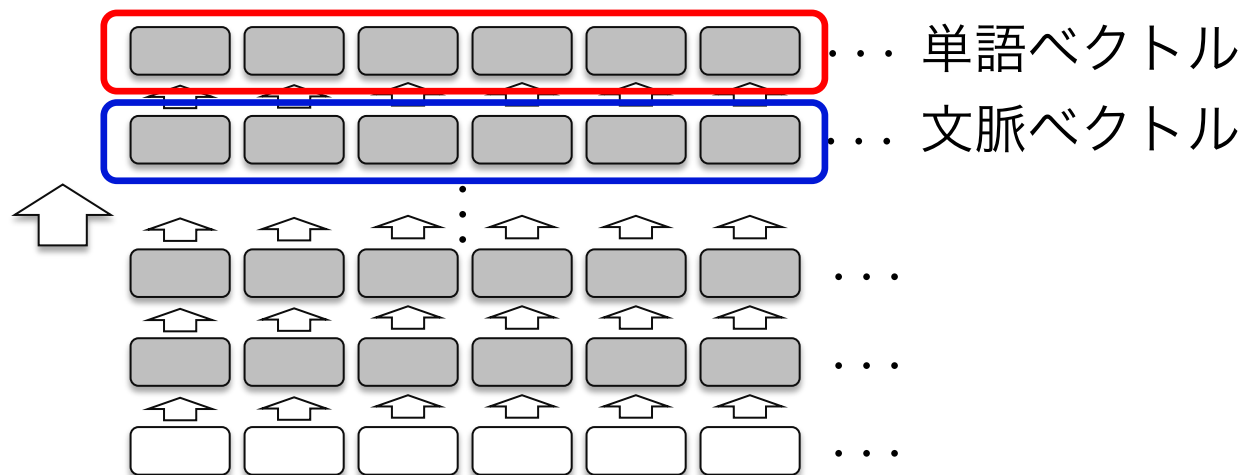
Transformerと単語ベクトル

- Transformerの出力層 $\approx$ 単語ベクトル

$$p(y|x) \propto \exp(\lambda(x)^T \gamma(y)) \text{ 単語ベクトル}$$

- $\lambda(x)$: 入力単語列 x の「埋め込み」
- $\gamma(y)$: 出力 y の単語ベクトル

出力y: gets very tired of sitting by ...



入力x: alice gets very tired of sitting ...

単語のベクトル化

- 単語ベクトルとは: 単語を実数のベクトルとして表現・処理するもの
 - 「単語埋め込み」 (embedding)ともいわれる
- ベクトル化しないと、“北陸鉄道” “ザンペリーニ”などに関する条件つき確率を個別に持つ必要
→ ほぼ不可能！
 - eg: 「ザンペリーニ→選手」 「北陸鉄道→沿線」などの確率を独立に計算するのは無意味
- LLMでは自動学習されるが、もともとは単語ベクトルを計算するWord2Vec (Mikolov+ 2013)が先に提案された

学習された単語ベクトル (一部)

幕下 [-0.18127 -0.16785 0.0039202 0.17829 -0.21384 0.094747
0.29718 0.5263 -0.30177 0.11019 0.16902 -0.30849 -0.19053 -0.11944
.. 0.093231 0.87569 0.025137 0.0011367 -0.025216 -0.12178 0.1811]

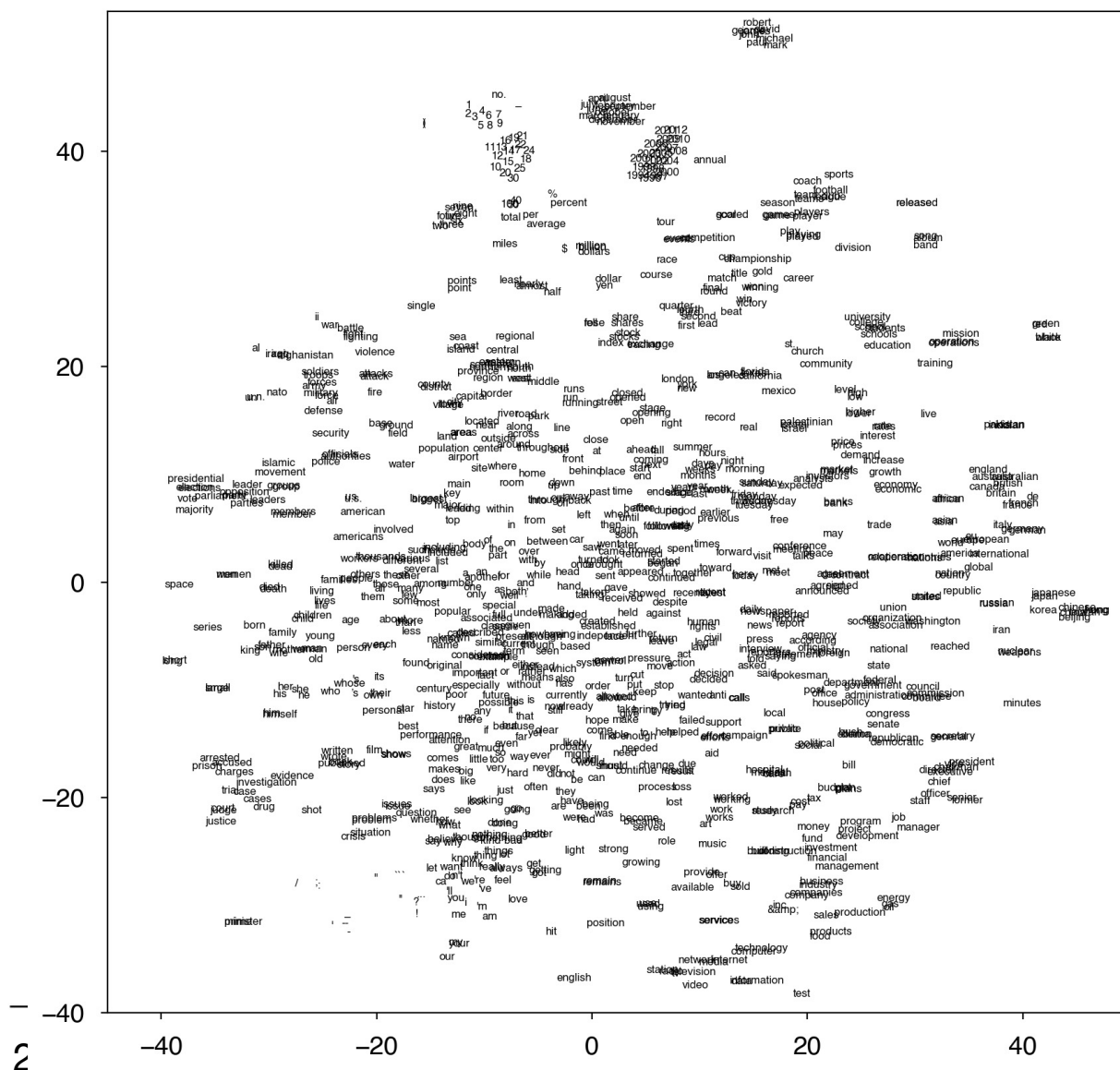
ディレクター [0.0015758 -0.15125 0.21817 -0.12623 0.13215 0.10187
0.080754 -0.024508 -0.048359 -0.13401 -0.14256 0.083109 0.30153
.. -0.077481 -0.15859 0.1278 0.028061 -0.15356 -0.11111 0.29463]

正義 [0.002604 0.00062024 -0.025644 -0.082722 0.077728 0.0087165
0.049547 -0.025976 -0.095092 -0.067749 0.013562 0.060091 0.053064]
.. 0.0228 -0.041253 0.1974 -0.10254 -0.029888 0.033918 0.21347]

静岡 [0.039938 0.10639 -0.021873 -0.014316 0.015339 0.044891
0.0024299 -0.0086346 -0.054358 -0.022956 -0.058226 -0.011501
.. -0.005193 -0.081363 -0.033958 -0.058588 -0.080491 0.09922]

高まる [0.036186 0.033845 -0.018044 -0.034166 -0.021112 0.0028494
0.12283 -0.090323 -0.029995 0.037449 -0.030411 0.014132 0.040336
.. 0.00908 0.0084737 -0.007582 -0.14347 0.029086 0.14277 0.11097]

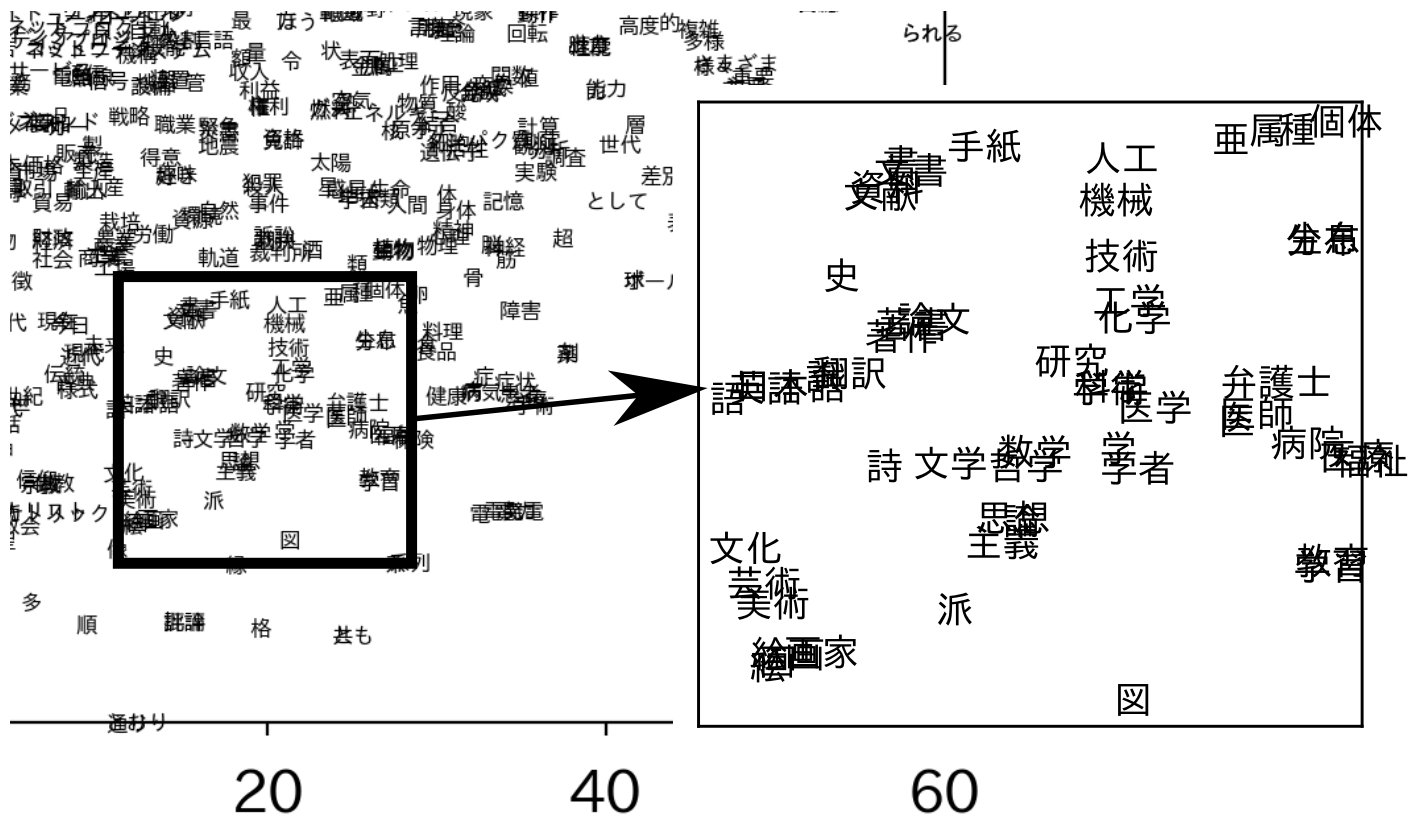
単語ベクトルの分布 (英語の場合)



- 300次元の単語ベクトルの上位1000語を、 t -SNEで2次元に描画
- 意味的なかたまりがみられる

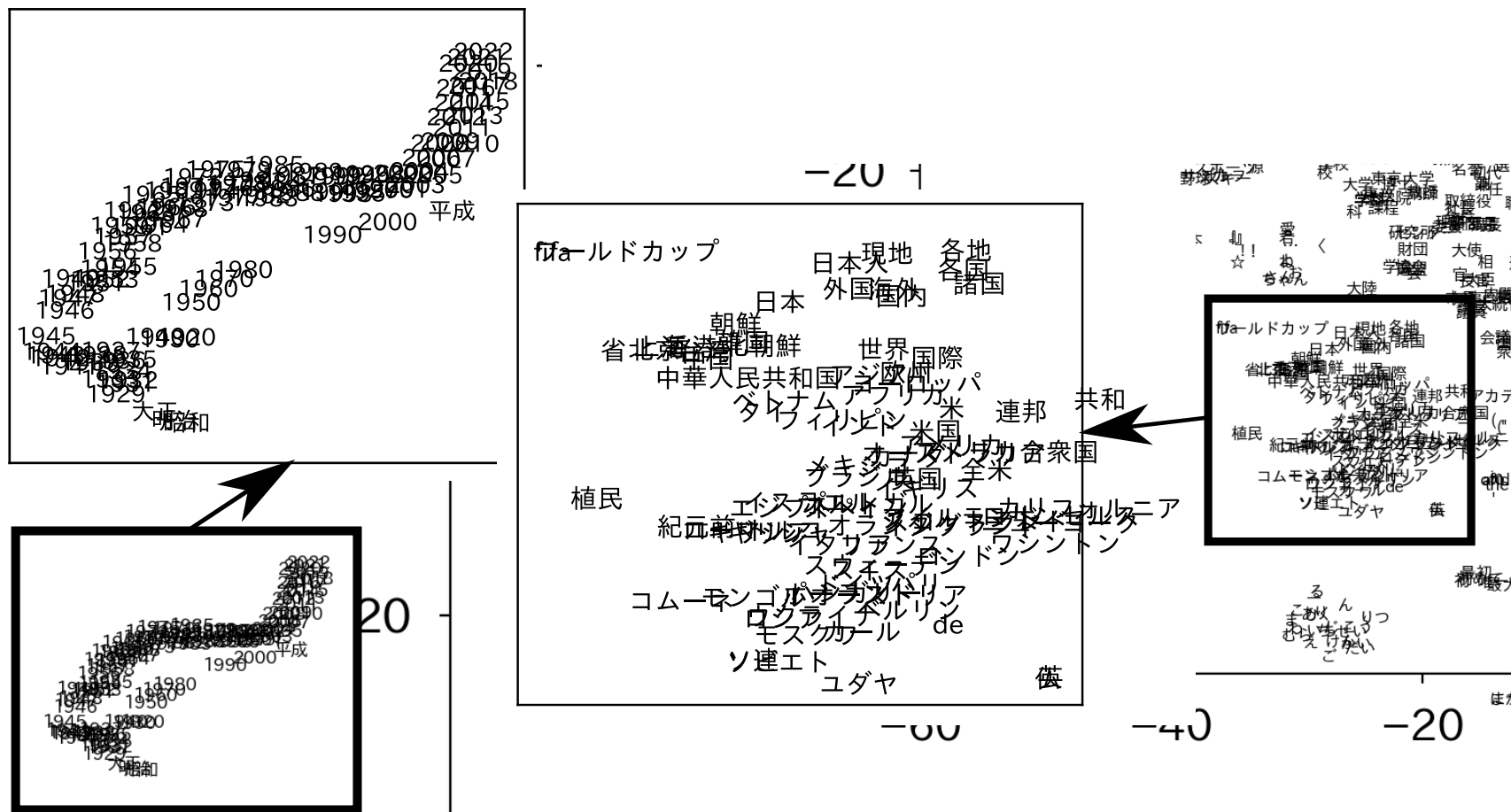
- Research Organization of Information and Systems
The Institute of Statistical Mathematics

日本語単語ベクトルの分布 (2)



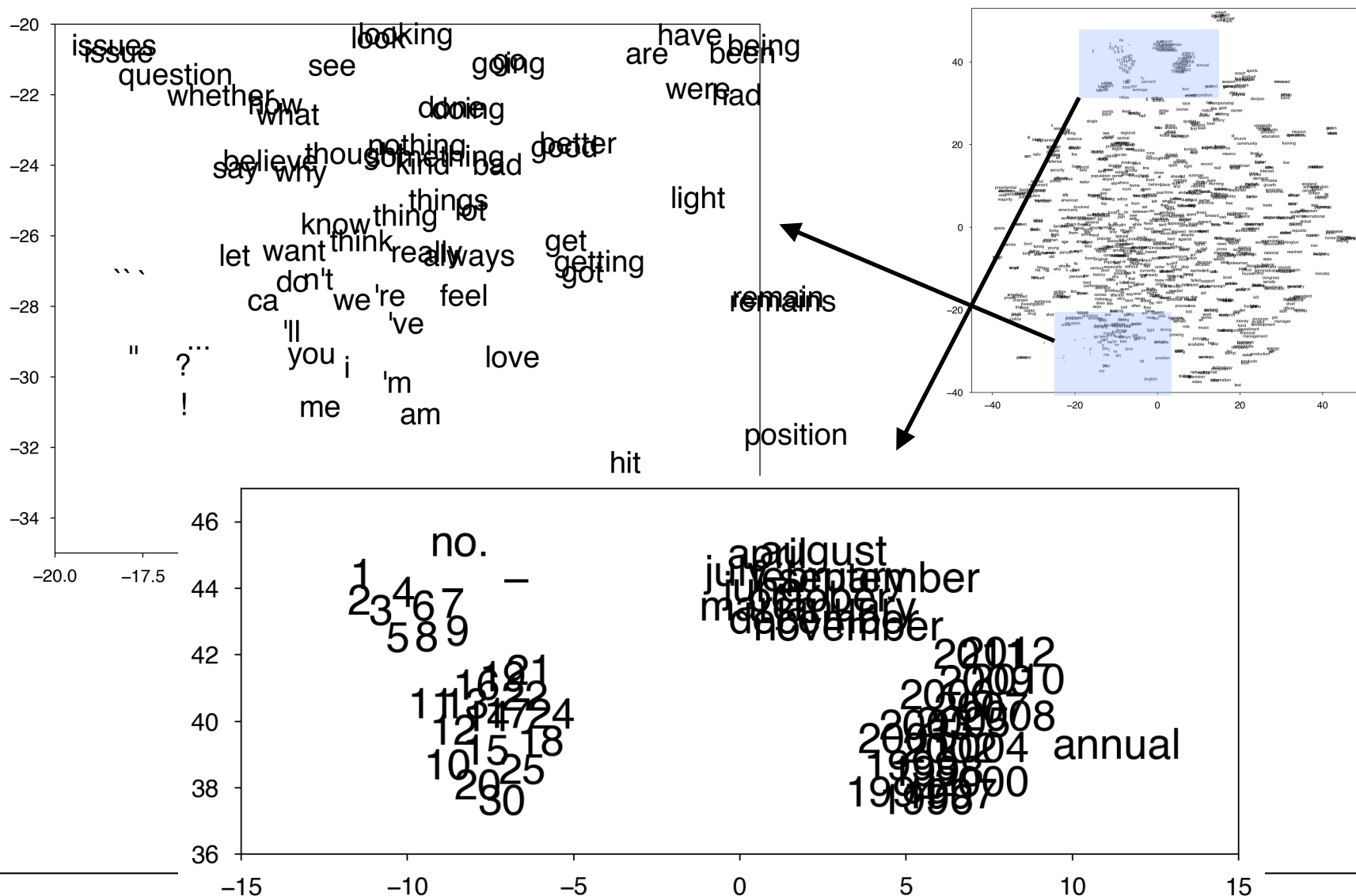
- 意味的に関連する単語が、自動的に近くに配置されている

日本語単語ベクトルの分布 (3)



- 西暦や年号、国名などがまとまって分布している

単語ベクトルの分布 (拡大図)



言葉の類似度の計算

- 単語ベクトルが似ていれば、意味が似ている
→ ベクトルの内積を計算すれば、意味の似ている単語がわかる

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + \cdots + x_K y_K$$

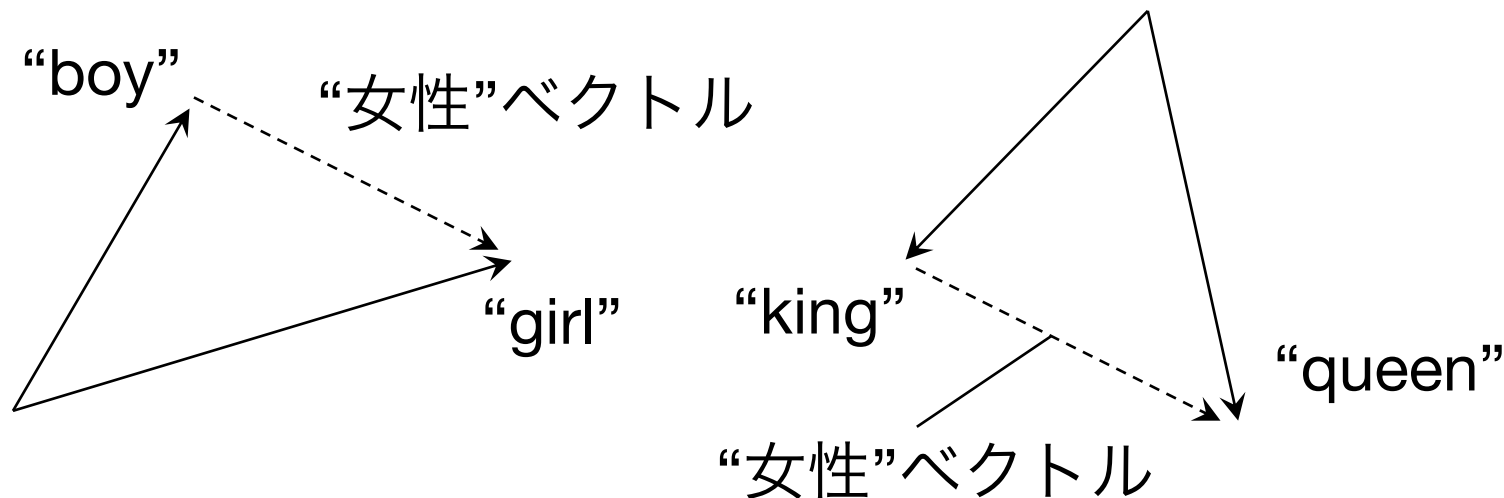
% similar.py model 静岡

静岡 -> 1.0000
滋賀 -> 0.8056
島根 -> 0.7873
岐阜 -> 0.7834
神奈川 -> 0.7770
大分 -> 0.7758
山梨 -> 0.7742
岡山 -> 0.7680
鳥取 -> 0.7622 ...

% similar.py model 正義

正義 -> 1.0000
幸福 -> 0.7800
唱え -> 0.7677
なき -> 0.7549
理想 -> 0.7538
われわれ -> 0.7431
信じる -> 0.7320
連帯 -> 0.7303
観 -> 0.7095 ...

言葉のベクトルの「計算」



- ベクトルの「引き算」をして別の言葉のベクトルに足すことで、言葉の意味の「計算」ができる！
- 言葉のベクトルの間で、

が成り立つ $\overrightarrow{\text{boy}} : \overrightarrow{\text{girl}} = \overrightarrow{\text{king}} : \overrightarrow{\text{queen}}$

意味の計算と認知科学

COGNITIVE PSYCHOLOGY 5, 1-28 (1973)

A Model for Analogical Reasoning¹

DAVID E. RUMELHART AND ADELE A. ABRAHAMSON

University of California, San Diego

A theory of analogical reasoning is proposed in which the elements of a set of concepts, e.g., animals, are represented as points in a multidimensional Euclidean space. Four elements A,B,C,D, are in an analogical relationship $A:B::C:D$ if the vector distance from A to B is the same as that from C to D. Given three elements A,B,C, an ideal solution point I for $A:B::C:I$

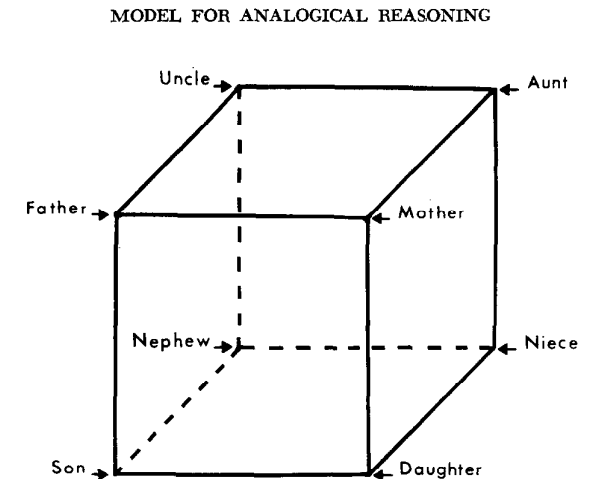
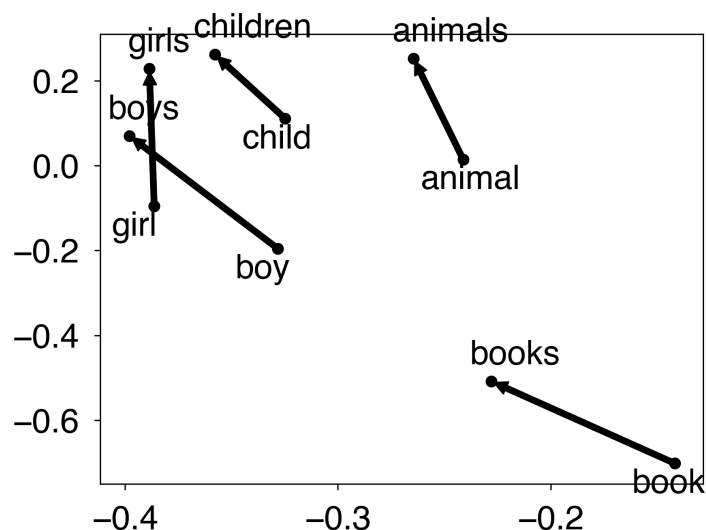


FIG. 2. Three-dimensional representation of relations among eight kinship terms.

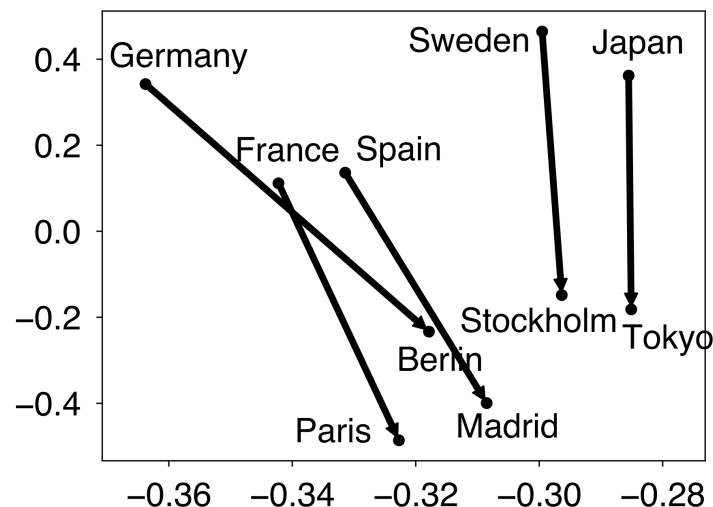
- 単語ベクトルの加減算の理論的分析を行っている
NeurIPS 2025の論文で、認知心理学の論文が
その基礎として引用されている!
(Rumelhart 1973)

言葉のベクトルの「計算」

- 実際のテキストから言葉のベクトルを作って計算してみたもの (持橋(2025)から抜粋)



“複数形”の関係



“首都”の関係

単語の「比例関係」の計算

- 単語ベクトルの間には、 $\vec{a} : \vec{b} = \vec{c} : \vec{d}$ の比例関係が成り立つことが知られている

```
contrast (vectors9,  
  "日本", "東京", "フランス")
```

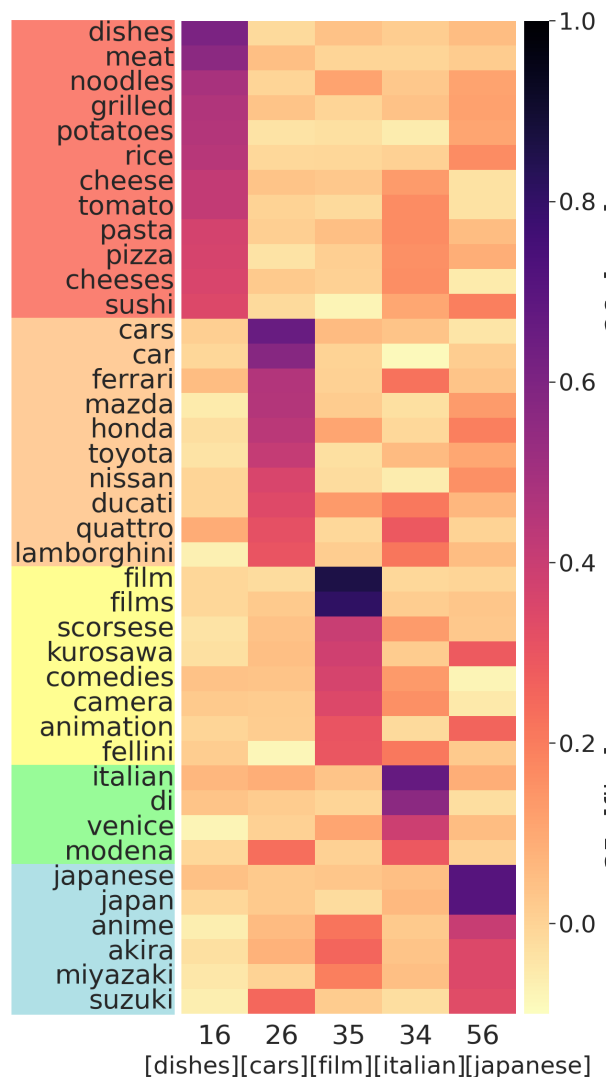
```
⇒ パリ -> 0.5812  
   リヨン -> 0.5394  
   マルセイユ -> 0.5321  
   トゥールーズ -> 0.5123  
   ニース -> 0.5098  
   ストラスブール -> 0.4985  
   ナント -> 0.4955  
   デイジョン -> 0.4722  
   ブリュッセル -> 0.4655  
   ボルドー -> 0.4468  
   マドリード -> 0.4367
```

```
contrast (vectors9,  
  "日本", "聖子", "アメリカ")
```

```
⇒ リンダ -> 0.4708  
   ジャネット -> 0.4679  
   マライア -> 0.4667  
   マリリン -> 0.4603  
   ジョニー -> 0.4508  
   ビリー -> 0.4478  
   テイラー -> 0.4402  
   ロジャー -> 0.4344  
   パウエル -> 0.4299  
   タウンゼント -> 0.4274  
   デイーン -> 0.4265
```

(「リンダ・ロンシュタット」は、
アメリカの松田聖子的な歌手)

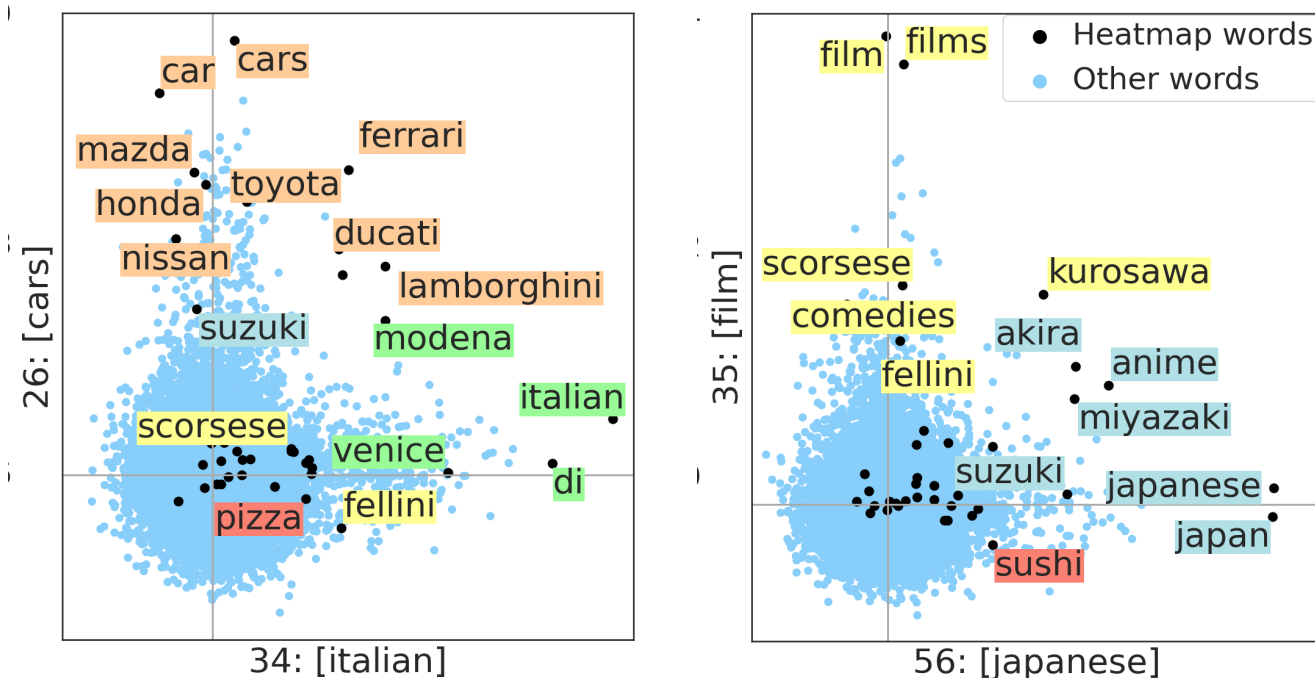
単語ベクトルのICAによる分析 [5]



- 単語ベクトル全体の埋め込み空間に独立成分分析(ICA)を行うことで、初めて単語ベクトルが複数の「解釈可能な軸」で説明できることを示した
- 京大下平研究室の仕事 (2021)



単語ベクトルのICAによる分析 [5]



- 左: “車” 軸と “イタリア” 軸の関係
- 右: “映画” 軸と “日本” 軸の関係
 - フェラーリや黒澤、アニメ、宮崎などに注目

最近の研究 (1)

- Sakata+ (arXiv 2026) : 階層的概念がLLMの内部でどう表現されているか？

Preprint. Under review.

Linear Representations of Hierarchical Concepts in Language Models

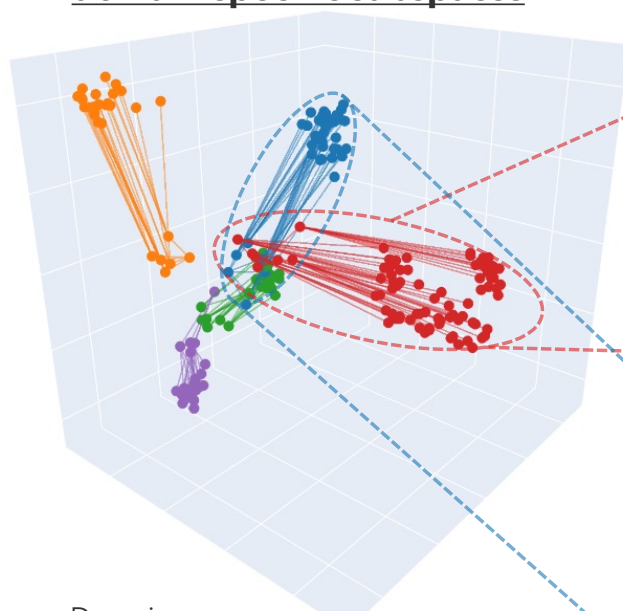
Masaki Sakata^{1,2}, Benjamin Heinzerling^{2,1}, Takumi Ito^{1,4},
Sho Yokoi^{3,1,2}, Kentaro Inui^{5,1,2}

¹ Tohoku University ² RIKEN ³ NINJAL ⁴ Langsmith Inc. ⁵ MBZUAI
sakata.masaki.s5@dc.tohoku.ac.jp benjamin.heinzerling@riken.jp
t-ito@tohoku.ac.jp yokoi@ninjal.ac.jp kentaro.inui@mbzuai.ac.ae

最近の研究 (2)

- Llama 3.1 8Bでの埋め込み表現の内部構造
→ 実際に階層構造になっている

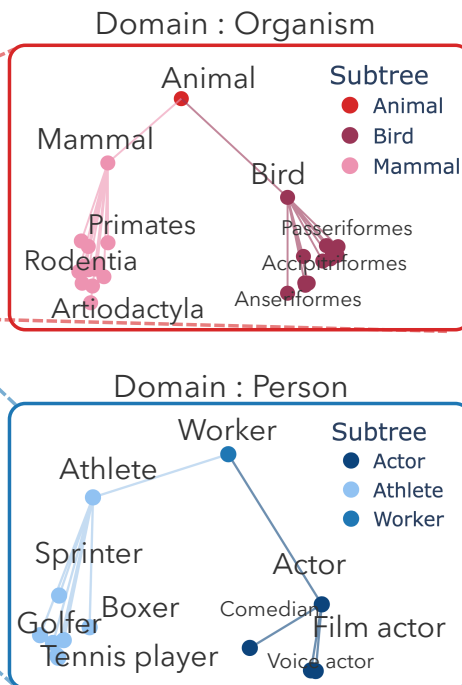
2. Hierarchical structure is encoded in domain-specific subspaces



Domain

- Location (orange)
- Person (blue)
- Organization (green)
- Organism (red)
- Research Topic (purple)

3. Hierarchical representations exhibit similar structure across domains



Further reading



- 『数理科学』
2025-10月号で、
機械学習における
幾何が特集されて
います
- 自然言語に関する
記事は、横井君
によるもの
(「言語の表現空間
の形」)

前半のまとめ

- 大規模言語モデル＝Transformerがどのように言語を理解しているかについては、一部で研究されているが、まだまだ謎が多い
 - － 単語ベクトルについては、統計的な解析も進められている
- 汎化誤差のような研究とは別
- 自然言語処理分野自体ではマイナー：機械学習の知識が必要なため
 - 研究の余地が多分にある！！

後半

確率的潜在意味スケールリング (PLSS)

はじめに



- テキストを連続的な尺度で測りたい場合は、多くある
 - アンケートがどれくらい肯定的か、否定的か
 - ホテルの評価で特に否定的(肯定的)なレビューはどれか
 - ある政治家の発言や法案がどれくらい保守的か/革新的か
 - 書簡から伺える作家の分裂症気質がどれくらいか

テキスト分類とは異なる!

- 1/0の単純なテキスト分類では対処できない
(自民党ならば右派なのは自明なので、その中の違いが重要)
- 人間が事前に付けたラベルではなく、自由な軸でテキストの連続的なスケールを測りたい
(英語は南北戦争前後でどのように変わったのか?)
- 教師データは一般に存在しない→教師なし学習

政治学方法論 (Political Methodology)

- 政治学に関する計量分析とその方法論の分野
 - 政治に関して、科学的・客観的な言明を行いたい
 - 文系分野としてはかなり高度な統計学・機械学習
- 特に、テキストの分析は中でも大きな分野
 - 背景：政治的なテキストのデータ化が進んでいる
(国会会議録、地方議会議事録、法案のテキスト、外交文書、共産党機関紙、...)
 - 2019年秋に、POLTEXT 2019に招待されて参加

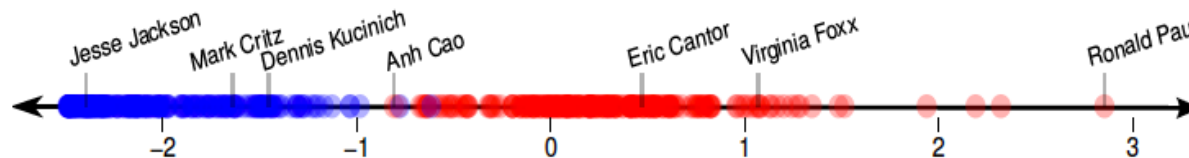
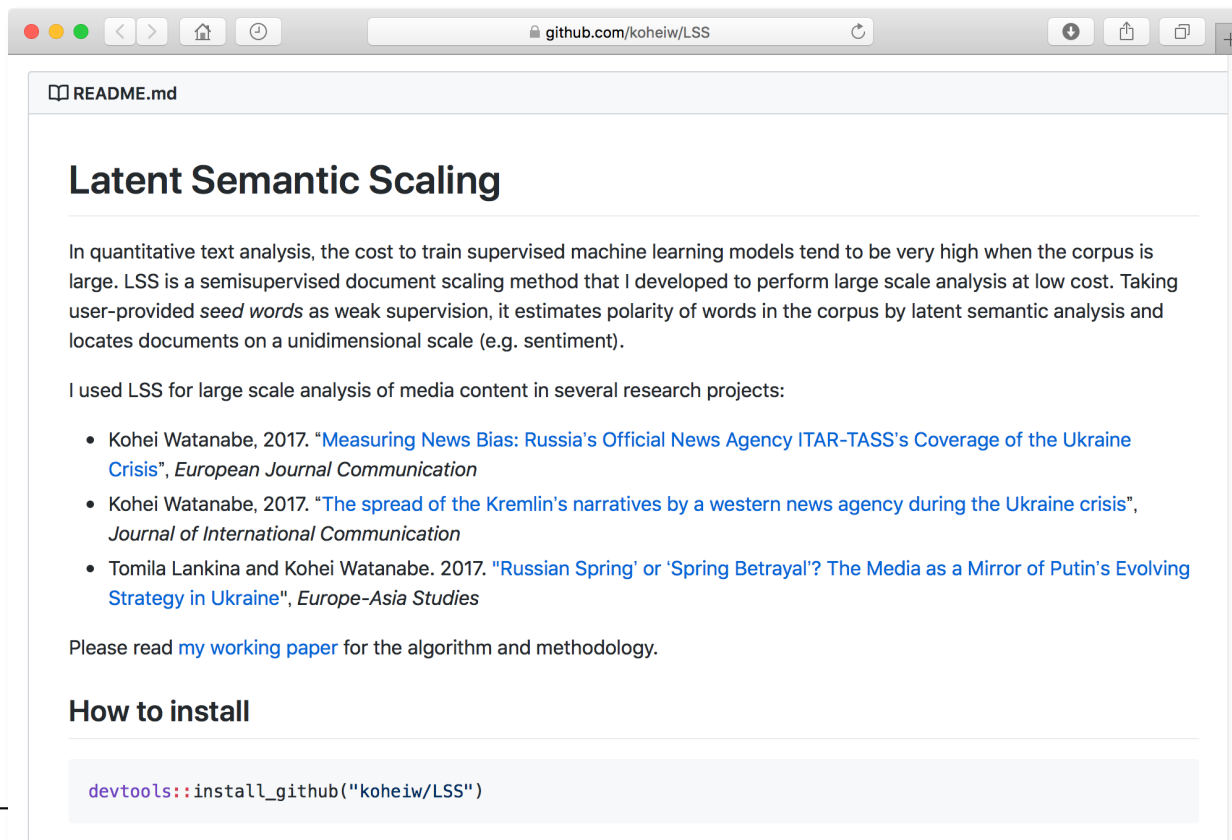


Figure 1: Traditional ideal points separate Republicans (red) from Democrats (blue).

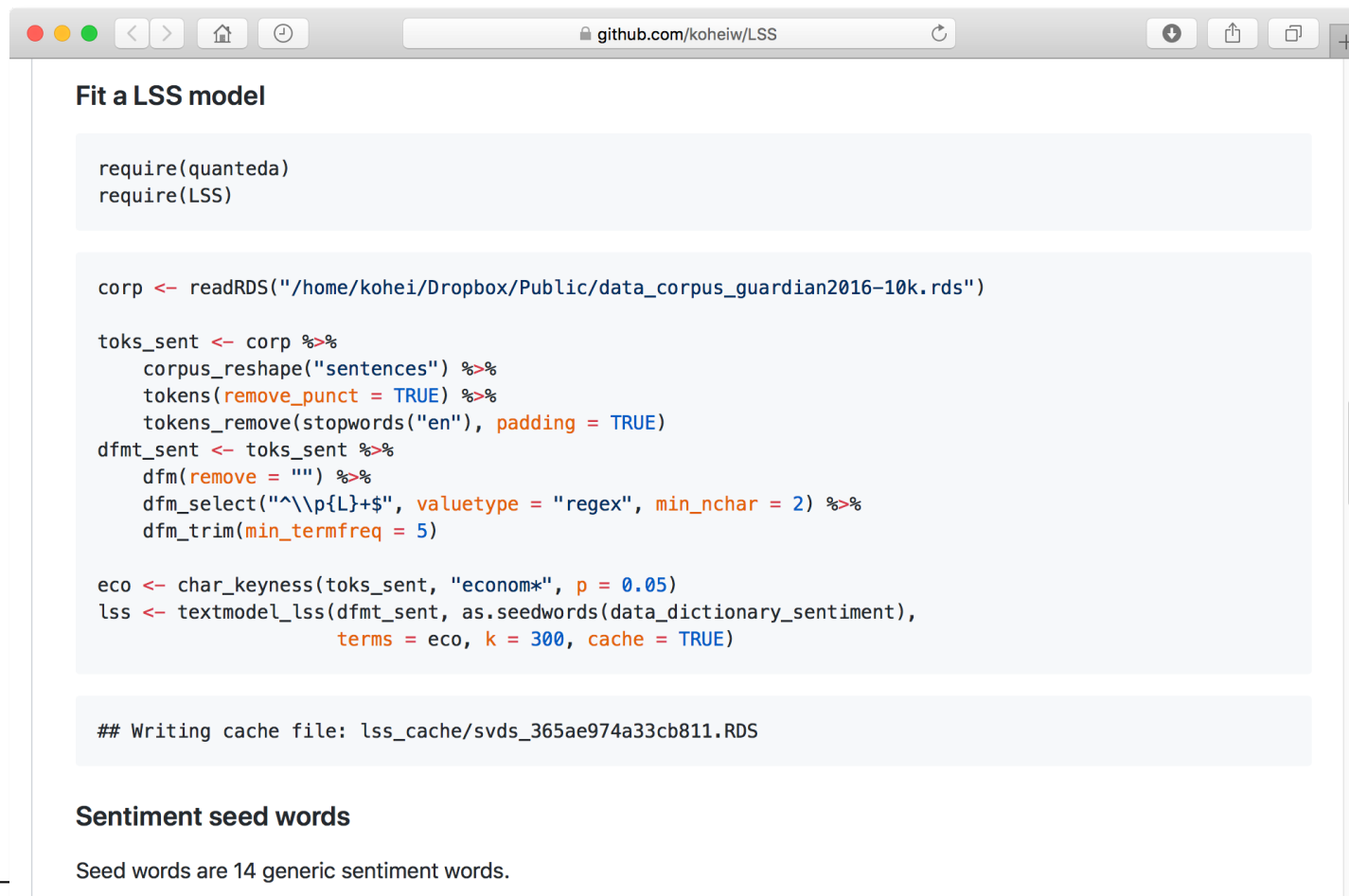
LSS (Latent Semantic Scaling)

- LSE(当時)の渡辺耕平氏が開発 (Watanabe 2015/2020)
- テキストをある視点で時系列分析するためのRパッケージ



LSS/quanteda

- Rを使って手軽に分析できるため、世界的に使われている



```
Fit a LSS model

require(quanteda)
require(LSS)

corp <- readRDS("/home/kohei/Dropbox/Public/data_corpus_guardian2016-10k.rds")

toks_sent <- corp %>%
  corpus_reshape("sentences") %>%
  tokens(remove_punct = TRUE) %>%
  tokens_remove(stopwords("en"), padding = TRUE)
dfmt_sent <- toks_sent %>%
  dfm(remove = "") %>%
  dfm_select("^\\p{L}+$", valuetype = "regex", min_nchar = 2) %>%
  dfm_trim(min_termfreq = 5)

eco <- char_keyness(toks_sent, "econom*", p = 0.05)
lss <- textmodel_lss(dfmt_sent, as.seedwords(data_dictionary_sentiment),
  terms = eco, k = 300, cache = TRUE)

## Writing cache file: lss_cache/svds_365ae974a33cb811.RDS
```

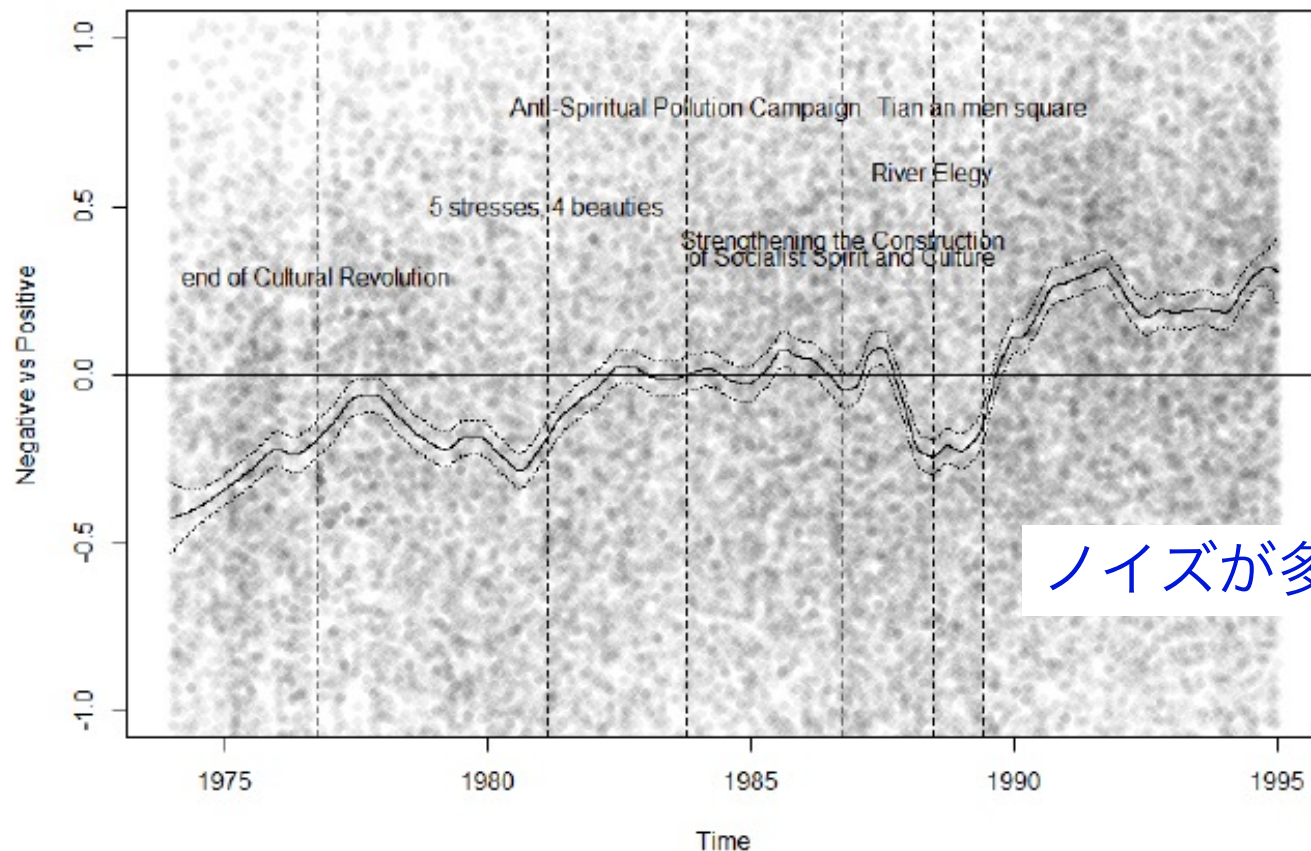
Sentiment seed words

Seed words are 14 generic sentiment words.

LSSによる分析例

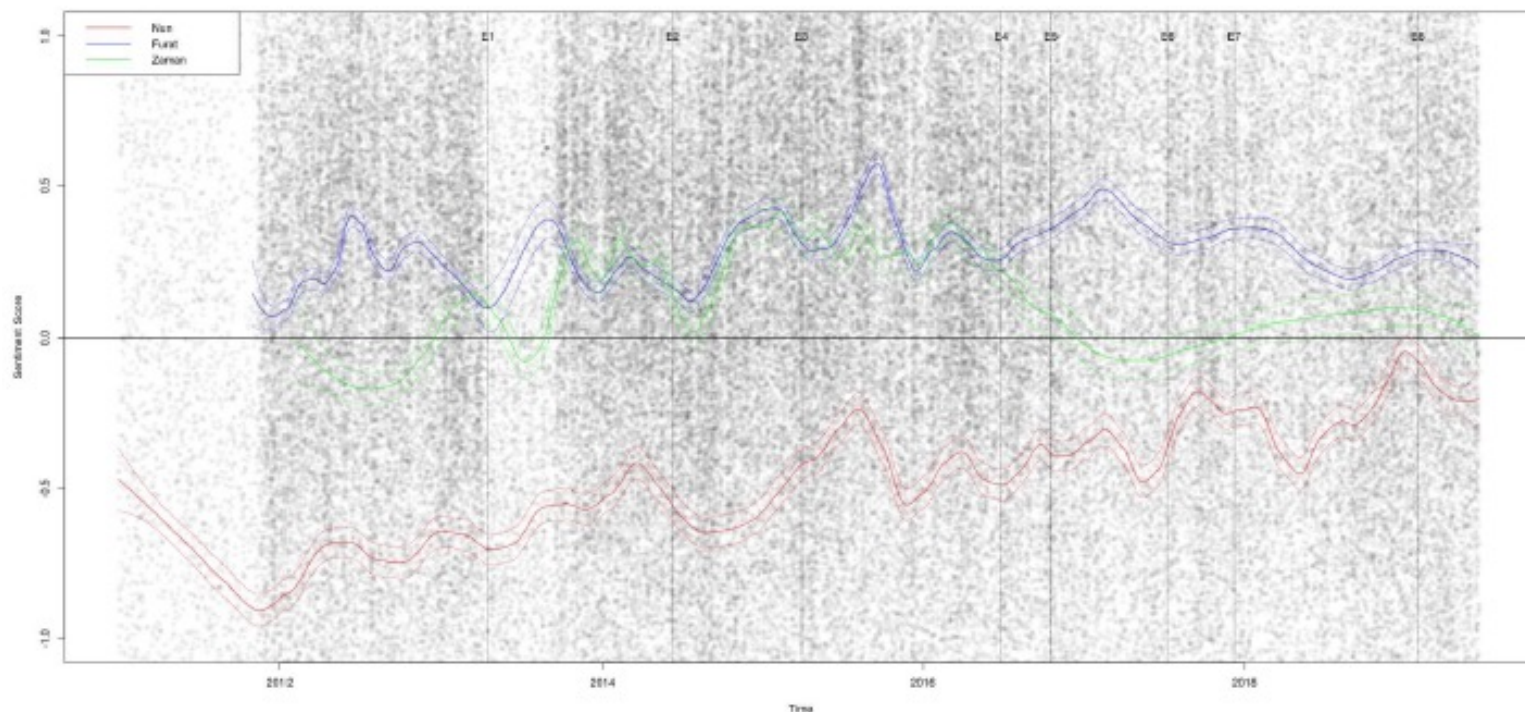
- 御器谷さん(慶応大学博士課程)による、中国の人民日報の中での「伝統」の使われ方の分析 @POLTEXT 2019

Figure 2 Valuation of tradition in the People's Daily



ノイズが多い！

LSSによる分析例 (2)



- 山尾さん (九大) / イスラム国の異なる宗派主義の時間変化, イラクの3種類の新聞記事(赤青緑) @POLTEXT
- 各文書での分散が非常に大きい

LSSの分析方法

- テキストの集合から、与えられた尺度の変化を抽出する方法 (古典的なベクトル空間モデル)
- 尺度：極性語とのベクトル空間での近さ (ex: positive-negative, left-right)

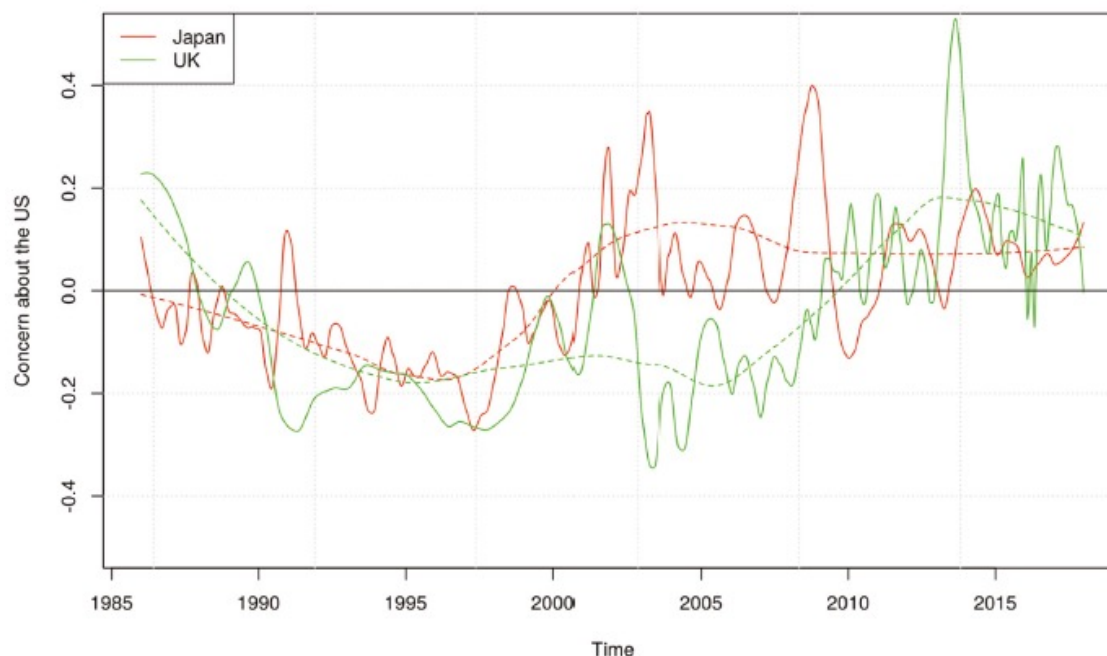


図2 日本とイギリスの新聞で見るアメリカに対する懸念

LSSによる分析方法 (1)

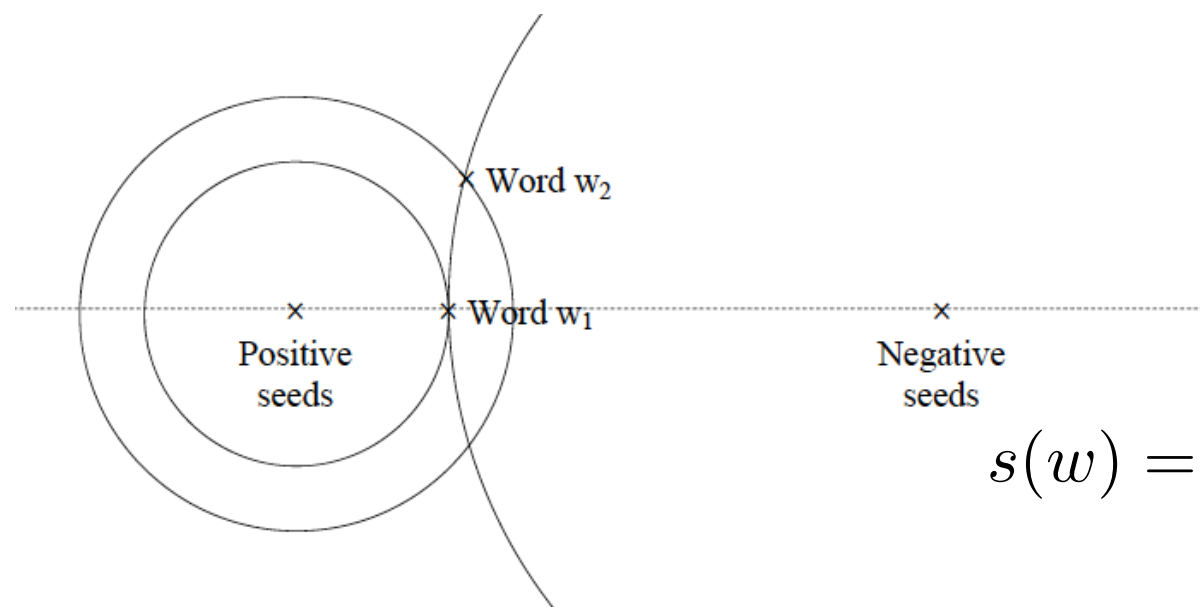
- 目標語 w との共起に基づいて、解析に用いる特徴語 c を選ぶ
- 例) 目標語 “econom*” との共起から選ばれた特徴語

Table 1: Entry words selected by econom*

Rank	Collocates	LL
1	growth	9969.0
2	social	5219.6
3	recovery	2264.4
4	political	2237.9
5	chief	1919.3
6	data	1518.5
7	inflation	1494.2
8	cycle	1462.6
9	rates	1346.8

LSSによる分析方法 (2)

- 特徴語を文書-単語行列の主成分分析で求めた、極性語とのコサイン距離でスコアリング



極性語 v_i は
複数あるので、
cos類似度を
平均

$$s(w) = \frac{1}{M} \sum_{i=1}^M \frac{\cos(\vec{w}, \vec{v}_i)}{\text{score}(v_i)}$$

Table 2: positive and negative economic words

Rank	Positive words		Negative words	
1	good	100.00	poor	-33.75
2	achieving	71.68	damage	-31.46
3	prospects	71.01	politics	-27.49
4	role	68.77	news	-26.39
5	successful	67.84	yen	-24.78
6	strategic	64.17	devalue	-24.08

Wordfish (Slapin and Proksch 2008)

- ポアソン確率モデルに基づく教師なしIRT
 - y_{ivt} を時間 t で政党 i が単語 v を使った頻度として、

$$p(Y) = \prod_t \prod_i \prod_v \text{Po}(\exp(\alpha_{it} + \psi_v + \beta_v * \theta_{it}))$$

を最大化する政党の位置 θ と単語パラメータ β を求める

- 完全な教師なし学習なので、得られる軸が分析したい軸と一致する保証がない
- 分析するテキスト集合に依存→選択の方法はない
- ポアソンモデルは、テキストの長さに依存

現在のテキスト分析の課題

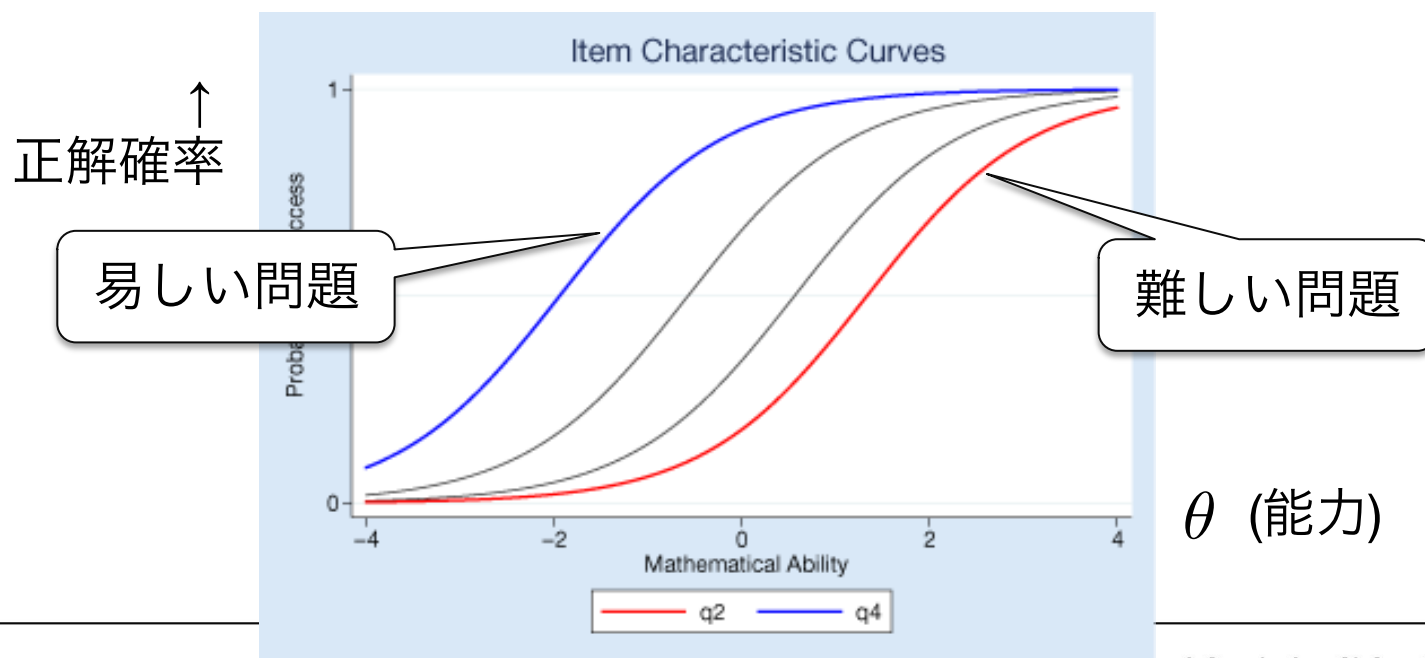
- LSS: 特徴語・種語の選択や前処理に人手が介在
 - ヒューリスティックにより、結果が変わってしまう
 - 統計的に何を計算しているのか、の基準が曖昧
- Wordfish: 抽出される軸が分析したい軸と一致している保証がない



Wordfishと項目反応理論(IRT)をベースに、LSSを
確率化したい…Probabilistic LSS (PLSS)

項目反応理論 (Item Response Theory)

- 心理統計学において、テスト結果から潜在的な能力を抽出するための方法 (eg. Lord (1952))
 - TOEICなどのテストの得点はIRTで計算されたもの
- 人ごとに、潜在的な能力 $\theta \sim \mathcal{N}(0, 1)$ が存在
→ ロジスティック回帰で問題への正解/不正解を観測



PLSSと項目反応理論 (IRT)

- SlapinがWordfish (ポアソンGLM) を提唱 (Slapin and Proksch 2008):

$$p(y_{ivt} = k) = \text{Po}(\exp(\alpha_{it} + \psi_v + \beta_v * \theta_{it}))$$

- この代わりに、次の多項モデルを考える
 - 多項分布 IRT:

$$\begin{aligned} p(v|\theta, \phi) &\propto p(v) \exp(\theta \cdot \phi_v) \\ &= \frac{\exp(\log p(v) + \theta \cdot \phi_v)}{\sum_{v=1}^V \exp(\log p(v) + \theta \cdot \phi_v)} \end{aligned}$$

- どんなモデル?

PLSS: Simple multinomial IRT

$$\begin{aligned} p(v|\theta, \phi) &\propto p(v) \exp(\theta \cdot \phi_v) \\ &= \frac{\exp(\log p(v) + \theta \cdot \phi_v)}{\sum_{v=1}^V \exp(\log p(v) + \theta \cdot \phi_v)} \end{aligned}$$

未知

未知

- テキストに、潜在的な極性 $\theta \sim \mathcal{N}(0, 1)$ が存在すると仮定
- 単語 v の確率は、ユニグラム確率 $p(v)$ に因子 $\exp(\theta \cdot \phi_v)$ が掛け合わされて決まる
 - θ と ϕ_v の正負が一致すると高い確率 $\rightarrow \phi_v$ は単語 v の極性
- θ も ϕ_v も未知の場合の教師なしロジスティック回帰と考えてもよい

PLSS: 単語ベクトルの利用

- 単語に関する言語的知識を使うために、単語ベクトルの利用を考える
- GloVeやWord2vec等で事前に計算された単語ベクトルを利用 (手元のテキストで計算してもよい)
- ニューラル単語ベクトルの空間に、以下で示すような部分空間が存在する: Ultradense Embedding (Rothe+ 2016)
 - DensRay (Dufter and Schutze 2019)は、以下の簡単な手法より明らかに性能が悪かった (詳しい議論は岩波本を参照)

PLSS: 単語ベクトルの利用 (2)

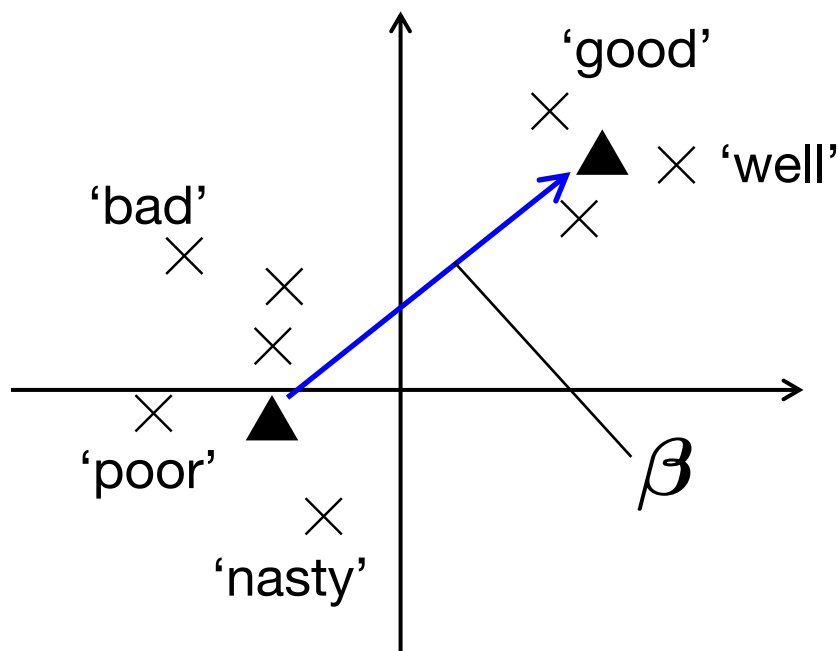
- 単語 v のベクトルを $\vec{v}$ と書くことにすると、単語毎のバイアス ϕ_v に代えて、次のモデルを考える
- 単語ベクトルに対する回帰係数 β を使って、

$$p(v|\theta, \beta) = \frac{\exp(\ell_v + \theta \cdot \beta^T \vec{v})}{\sum_v \exp(\ell_v + \theta \cdot \beta^T \vec{v})}$$

- β と $\vec{v}$ の内積 $\beta^T \vec{v}$ が、 ϕ_v と同じ働き
- LSSで使われているコサイン類似度の確率モデル化
- 単語ベクトルと同じ次元のパラメータ β を、1つ推定すればよい

β の推定

- 最も簡単には、 $+$ と $-$ を与えた少数の極性語辞書の単語ベクトルの平均の差を計算



- β のノルムは1にする→極性語として極端な単語を与える必要がない (LSSとの違い)

β の推定

- 各単語ベクトル $\vec{v}$ について、 $\phi_v = \beta^T \vec{v}$ が軸 β に沿った「単語の極性」を表す
- 例: 英語の一般的なpositive-negative辞書を使った場合

general positive-negative words, same as LSS

positive	good nice excellent positive fortunate correct superior
negative	bad nasty poor negative unfortunate wrong inferior

計算された単語の極性 $\phi_v = \beta^T \vec{v}$ (上位)

honored	0.4658	guest	0.3765	unique	0.3533
selected	0.4635	custom	0.3731	dedicated	0.3530
coveted	0.4254	suites	0.3719	lounge	0.3468
suite	0.4189	designing	0.3716	certified	0.3459
invited	0.4185	terrific	0.3708	wines	0.3451
excellent	0.4183	booked	0.3686	via	0.3447
tasting	0.4146	flexible	0.3654	elegant	0.3414
craft	0.4117	spa	0.3605	installed	0.3403
chosen	0.4116	wonderful	0.3579	happy	0.3402
nice	0.4053	slated	0.3574	venue	0.3400
designed	0.3884	plus	0.3558	prepared	0.3393
available	0.3858	buick	0.3554	design	0.3383
pleasant	0.3849	indoor	0.3552	preparing	0.3370
fortunate	0.3780	accessible	0.3544	fine	0.3369

計算された単語の極性 $\phi_v = \beta^T \vec{v}$ (下位)

greed	-0.5244	anger	-0.4404	rumors	-0.4045
rampant	-0.5135	scandals	-0.4394	backlash	-0.4031
panic	-0.4923	violence	-0.4367	reckless	-0.4018
fear	-0.4839	catastrophe	-0.4350	misery	-0.4018
racism	-0.4753	rage	-0.4317	enron	-0.4013
ignoring	-0.4693	violent	-0.4298	tide	-0.4010
unrest	-0.4618	abuses	-0.4234	depression	-0.3978
chaos	-0.4593	outrage	-0.4233	badly	-0.3972
intimidation	-0.4500	distress	-0.4202	confronting	-0.396
riots	-0.4532	abruptly	-0.4175	epidemic	-0.3940
subprime	-0.4514	blame	-0.4164	persistent	-0.3935
fears	-0.4489	bust	-0.4096	meltdown	-0.3921
collapse	-0.4429	resentment	-0.4065	deepening	-0.3886
fearful	-0.4418	anxiety	-0.4053	rhetoric	-0.3882

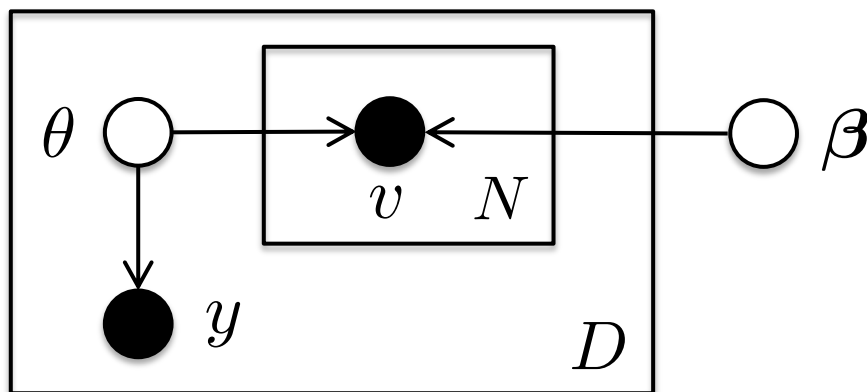
- 辞書でごく一部の単語を指定しただけで、多数の単語に適切な極性が与えられている！

β の半教師あり学習

- 辞書が分析前に **自明に準備できる**とは限らない
→ 辞書の代わりに、「正例のテキスト集合」
「負例のテキスト集合」を少数だけ与える
- 辞書と同時に使うことも可能：辞書による β を初期値にして学習
- データ：
 X_ℓ, Y_ℓ : ラベルがある少数のテキストとラベル
 X_u : ラベルのない多数のテキスト
— 機械学習的には、半教師あり学習の枠組み

β の半教師あり学習 (2)

- $p(Y_\ell|X_\ell)$ (だけ)を最大化するように β を学習



$$p(y, d|\theta) = p(y|\theta) \prod_{v \in d} p(v|\theta, \beta)$$

未知

$$p(v|\theta, \beta) = \frac{\exp(\ell_v + \theta \cdot \beta^T \vec{v})}{\sum_{v=1}^V \exp(\ell_v + \theta \cdot \beta^T \vec{v})}$$

未知

- 問題：ラベル付きデータの θ はそもそも未知！

β の学習 (1)

- θ を積分消去して、 β のみを学習する

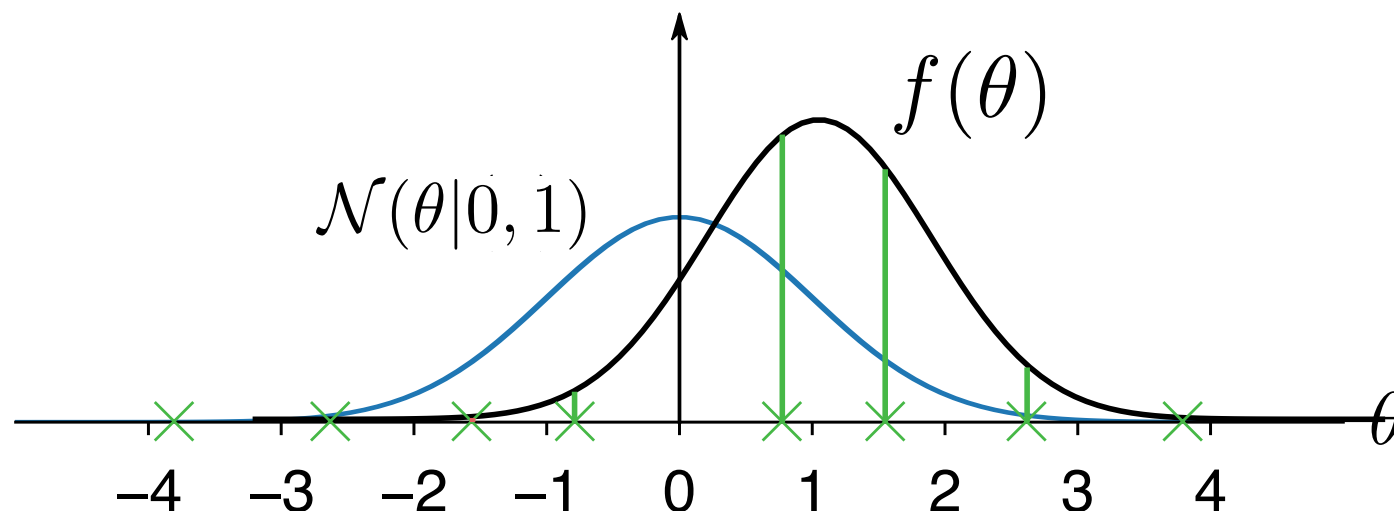
$$p(y, d|\beta) = \int_{-\infty}^{\infty} p(y, d, \theta|\beta) d\theta = \int_{-\infty}^{\infty} p(y|\theta) p(d|\theta, \beta) p(\theta) d\theta$$

- $p(\theta)$ は標準正規分布 $\mathcal{N}(0, 1)$ なので、Gauss-Hermite 求積で上の式は高精度に数値積分できる

$$\int_{-\infty}^{\infty} f(\theta) \mathcal{N}(\theta|0, 1) d\theta \simeq \frac{1}{\sqrt{\pi}} \sum_{i=1}^H w_i f(\sqrt{2}x_i)$$

- H 個の分点 $x_1 \cdots x_H$ とその重み $w_1 \cdots w_H$ は直交エルミート多項式から事前に数学的に求まる
- $f(\theta)$ を直交多項式で近似していることに相当

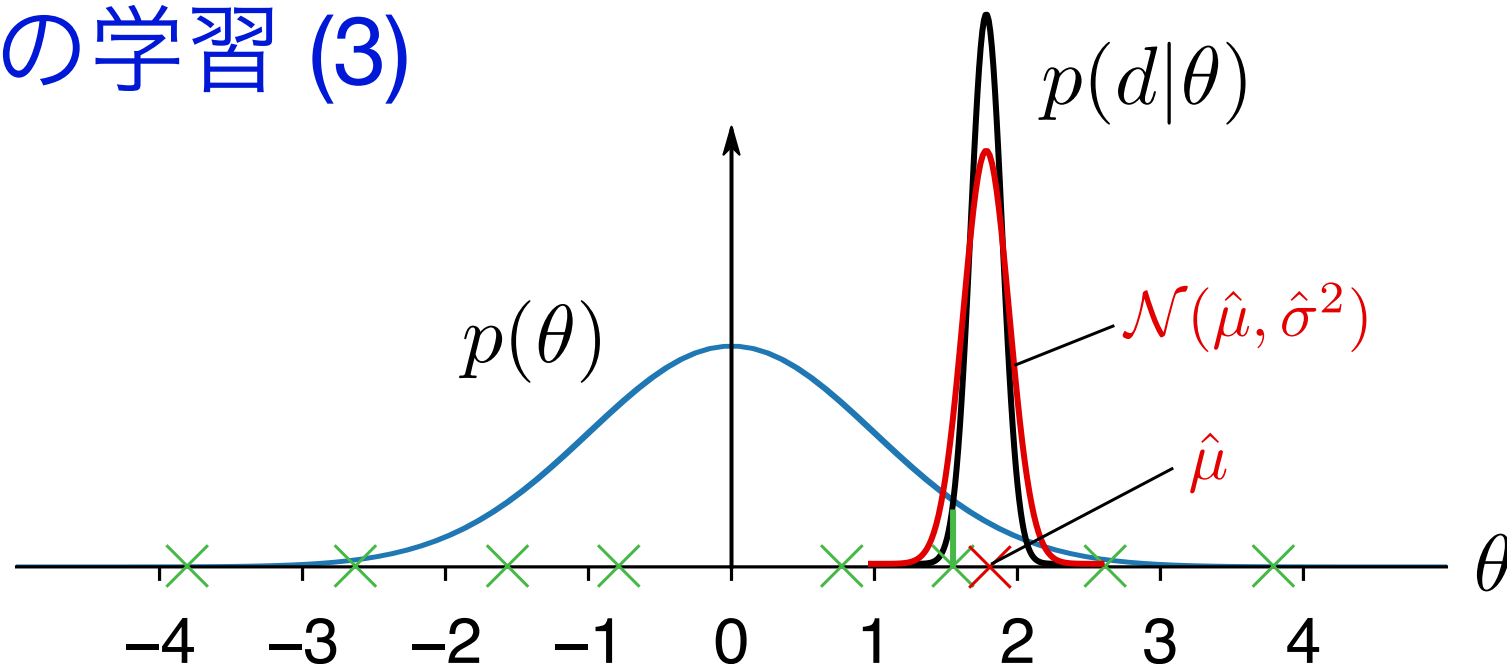
β の学習 (2)



- **Gauss-Hermite求積**: 緑の分点で計算するだけで、 $\mathcal{N}(0, 1)$ に関する積分は高精度に計算できる

$$\int_{-\infty}^{\infty} f(\theta) \mathcal{N}(\theta|0, 1) d\theta \simeq \frac{1}{\sqrt{\pi}} \sum_{i=1}^H w_i f(\sqrt{2}x_i)$$

β の学習 (3)



- 問題: 実際のテキストでは、尤度 $p(d|\theta)$ は特定の θ の周りに集中→緑の分点がヒットしない
- 解決策: 積分を仮想的に $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ で行うように変数変換 (Liu+ 1994, Biometrika)
 - $\hat{\mu}$ は2分探索で、 $\hat{\sigma}^2$ は数値二階差分で計算できる

β の学習 (4)

$$\log \int_{-\infty}^{\infty} e^{\ell(\theta)} \mathcal{N}(\theta|0, 1) d\theta$$

β を含む

$$\simeq \log \left[\frac{\sigma}{\sqrt{\pi}} \sum_{k=1}^K \exp \left\{ \log w_k + \ell(y_k) + \frac{1}{2} \left(\frac{1}{\sigma^2} (y_k - \mu)^2 - y_k^2 \right) \right\} \right]$$

- 実際の計算は、上式のようにすべて対数で行う
- β は K 次元のベクトルなので、さらにこの式を β で偏微分して勾配を計算 → L-BFGS で最適化
 - Hamiltonian MCMC によるベイズ推定も試したが、最適化による MAP 推定の結果がよかった
- ベクトル化による R の実装が大変。。

実験とデータ

- Young and Soroka (*Political Communication*, 2012) のテキストに対する人手のコーディングデータ

<doc>

The rate at which Americans bought new single family homes dropped 14 percent in September from the month before x the Government reported today . Resurging mortgage rates and a sharp jump in home prices apparently deterred potential ...

</doc>

EXCEPT for “conservative ,” which is the way Ronald Reagan describes himself x economic labels are difficult to pin on the President elect . One of the best ways to predict what he may do as President is to examine ...

- [“Neutral” “Negative” “Negative”], [“Neutral” “Neutral” “Neutral”] のような3人の評定を、5段階に変換
 - この生評定データを使った方がよい？

極性語

- 極性語の辞書としては、LSS (Watanabe 2020)で使われている次の標準的なpositive-negative辞書を使用

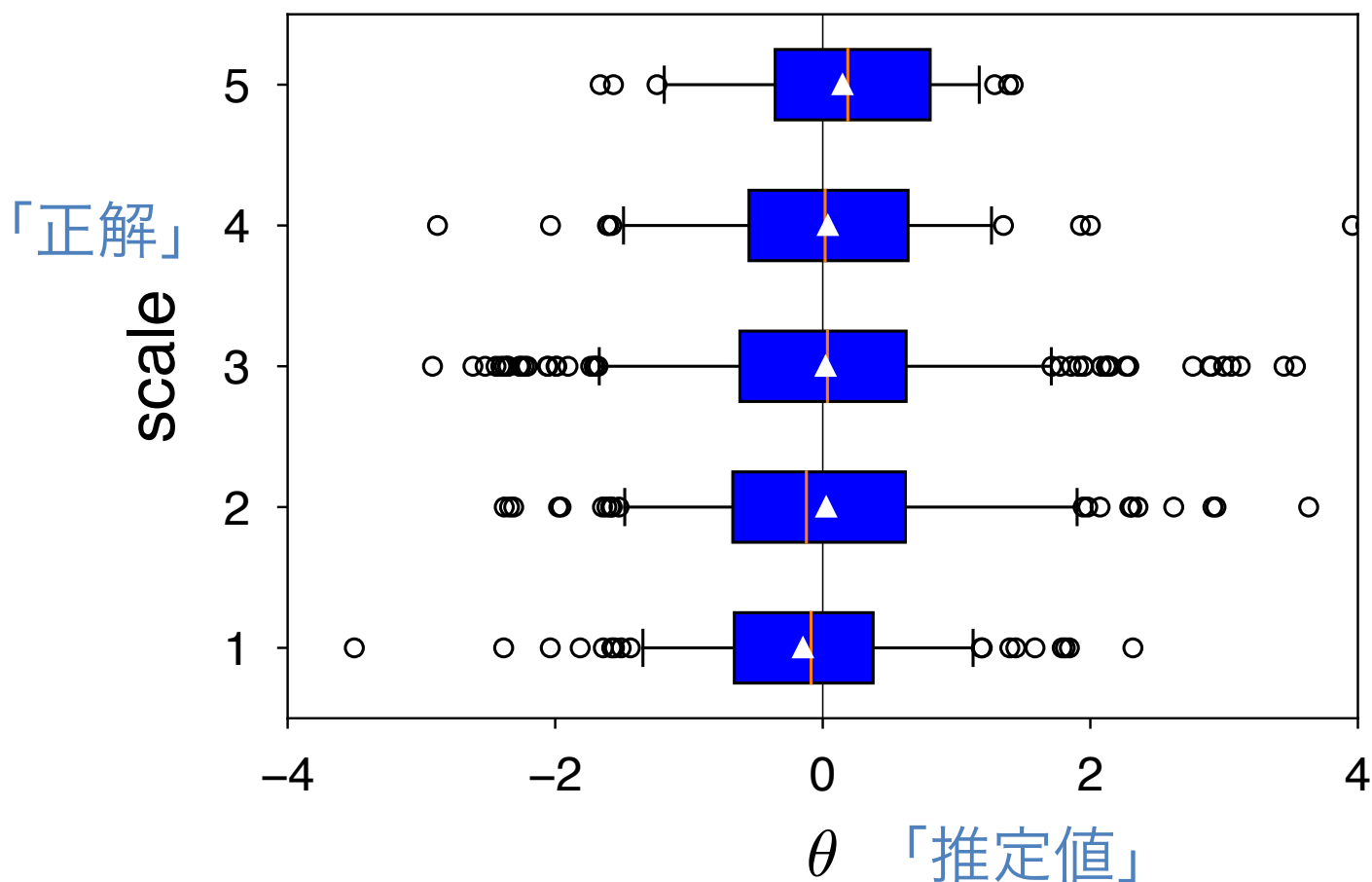
```
# LSS's data_dictionary_sentiment
```

```
positive      good nice excellent positive fortunate correct superior
```

```
negative      bad nasty poor negative unfortunate wrong inferior
```

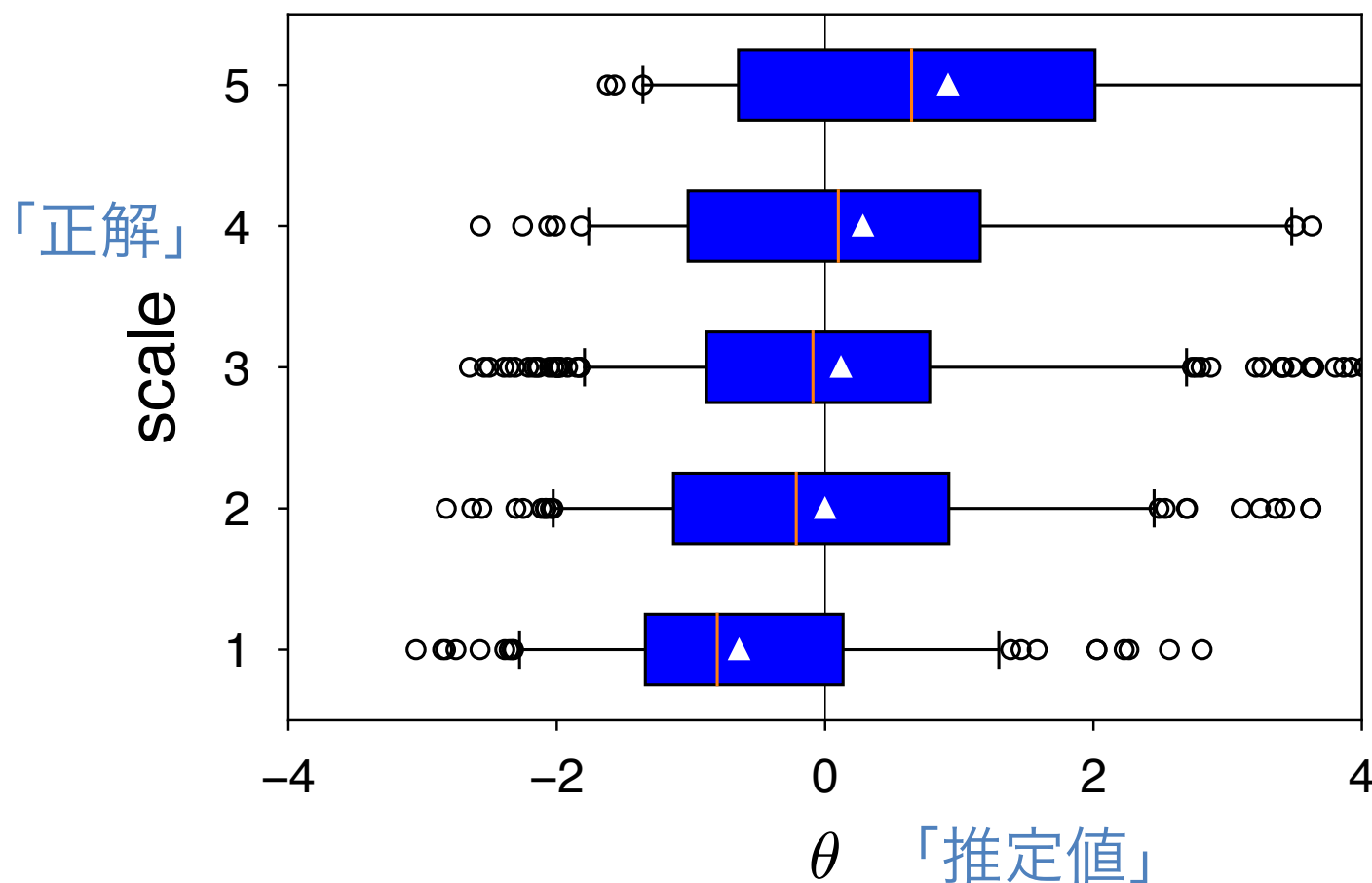
- 注: 提案法はPositive-Negativeのような自明な軸だけでなく、任意の複雑な意味的な軸を扱える

LSSと人手のスケールとの相関



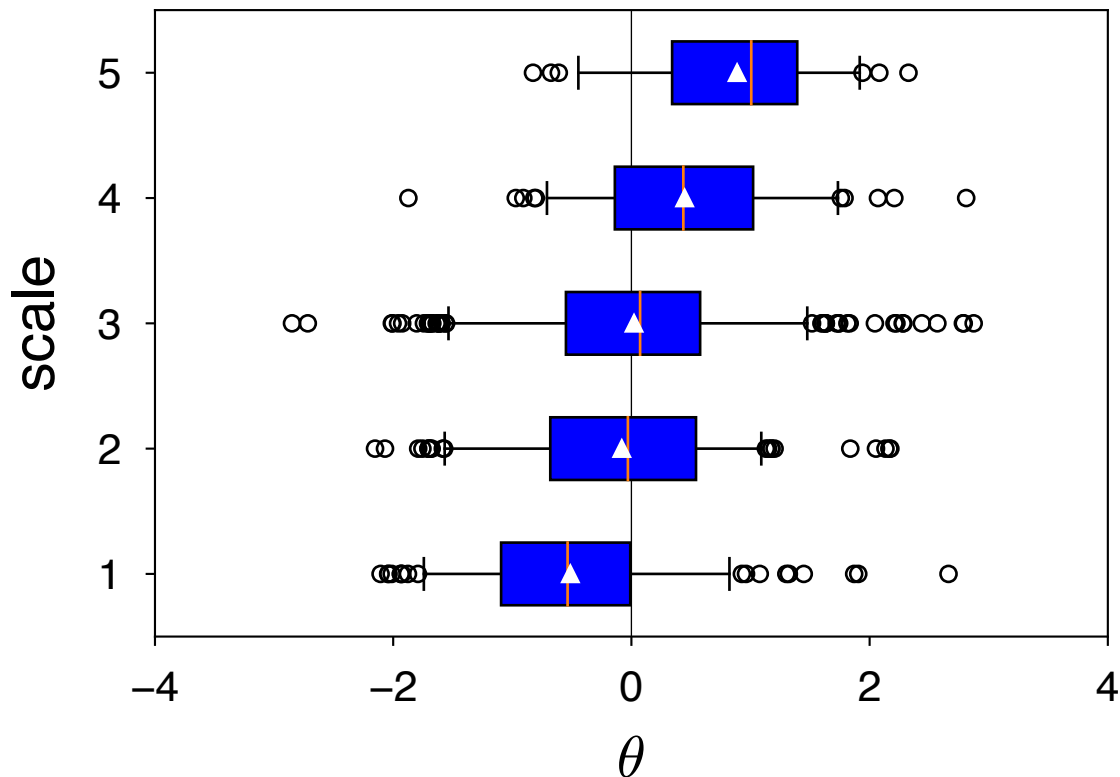
- 平均値($\triangle$)は微妙にスケールに相関しているが、非常にノイズが多い状態 (相関係数 $\rho=0.065$ 、平均だと0.901)

PSSと人手のスケールとの相関



- LSSよりかなり高い相関が出る ($\rho=0.240$, 平均だと0.965)
- モデルが Bag of words なので、さらに改善の余地あり

極性辞書＋半教師あり学習



- 標準的な極性辞書に正例/負例を15件加えて学習
- 相関係数 $\rho=0.348$, 平均(Δ)を使えば0.985に上昇

国会議事録の分析

- 2021年春の通常国会の衆議院・農林水産委員会の議事録を使用 (Webからスクレーピング)
 - 議員の論点の違いが比較的明らかなため
 - 12回の開催、1,324個の発言
- 例として、玉木雄一郎議員(国民民主党)の発言を+、田村貴昭議員(日本共産党)の発言を-として、それぞれランダムに20発言を抽出して半教師ありテキストとする
- β を最適化して求め、他の発言をPLSSで分析して潜在的な極性 θ を計算
- 極性辞書は使わない (準備できない)

国会議事録の分析 (結果)

θ	発言者	発言内容
1.6410	大串（博）委員	立憲民主党・無所属の大串です。早速質疑に入ります。貯保法ですけれども、私は、
1.4988	矢上委員	時間の関係で次の質問に移らせてもらいますけれども、低コスト化対策ですね。二問あつ
1.4838	重徳委員	だから、農水省に何の非もないのかと言っているんですよ。例えば、大臣、富山県の御
1.3139	大串（博）委員	全くちぐはぐですね。一時的な要因で余っているんだったら、一時的に市場から切り離さ
1.1895	本郷政府参考人	木材流通に関してでございます。需給のミスマッチを起こさないように、生産、加工の事
1.0874	玉木委員	国民民主党の玉木雄一郎です。本法案についてまず質問いたします。先ほどから、規
1.0784	玉木委員	コロナにはいろいろなことを教えてもらったなと思ったんですが、例えば、マスク一つ国
1.0716	近藤（和）委員	石川県能登半島の近藤和也でございます。よろしく願いいたします。COVID-1
1.0316	金子（恵）委員	今、イノベーションの話もされたので、済みません、順番を変えて、林業労働力の育成、
0.8315	神谷（裕）委員	そうしますと、遡れる限り遡るといことだと思ふんですが、そこで、先ほど議論になっ
:		
-0.5228	野上国務大臣	御指摘のございました主要農作物種子法につきましては、昭和二十七年に、戦後の食料増
-0.7074	野上国務大臣	間伐等特措法によりまして、平成二十年の法律制定後、一定以上の森林面積を有します市
-0.8432	葉梨副大臣	お答えいたします。佐々木先生の資料の二の品目横断的経営安定対策、これが導入され
-1.0359	水田政府参考人	お答えいたします。委員御指摘の冊子の二ページのところに「EUやアメリカの現状」
-1.5408	高鳥委員長	お諮りいたします。ただいま議決いたしました法律案に関する委員会報告書の作成につ
-1.5771	高鳥委員長	起立少数。よって、本修正案は否決されました。次に、原案について採決いたします。
-1.9059	大串（博）委員	今回の農中さんの議論を契機に是非いい議論をしていただきたいと思いますし、間違っ
-2.2223	田村（貴）委員	私は、日本共産党を代表して、本法案に反対の立場から討論を行います。第一に、改正
-2.5166	田村（貴）委員	私は、日本共産党を代表して、畜舎等の建築及び利用の特例に関する法律案に反対の討論
-2.6210	重徳委員	立憲民主党の重徳和彦です。今日は矢上筆頭、先輩、同僚議員の御了解をいただきまし
-3.5215	新井政府参考人	お答えいたします。OIE連絡協議会は、産業界及び学界における技術者又は学識経験

● 玉木－田村の極性軸上に、連続的に発言を並べられる

国会議事録の分析 (結果)

- 玉木－田村議員を極性軸とした際の、各単語の極性

$$\phi_v = \beta^T \vec{v}$$

(玉木側)		(田村側)	
v	ϕ_v	v	ϕ_v
まずは	0.5167	訴訟	-0.5769
なので	0.4560	傍聴	-0.5646
一つ	0.4364	毀損	-0.5519
ミリ	0.4246	原告	-0.5481
増やす	0.4178	敗訴	-0.5147
整い	0.4109	控訴	-0.5010
もっと	0.4014	係争	-0.4963
もう少し	0.4012	裁判所	-0.4962
植え付ける	0.3995	判決	-0.4808
切り替える	0.3985	審	-0.4727
一番	0.3982	シベリア	-0.4599
どうにか	0.3940	退け	-0.4491
しっかり	0.3903	高裁	-0.4462
同時に	0.3897	弁護	-0.4431
早く	0.3890	最高裁	-0.4410
戦える	0.3886	紛争	-0.4303

Future Work

- Transformerの内部状態を利用した拡張
 - 実際のテキストには“not”などの反転表現や反実仮想などが含まれるため、Transformerで精密に見ることがおそらく有効
- Benoit+ (2016)でのクラウドソーシングされた評価データでの実験
- 英語論文、Rスクリプトのパッケージ化

後半のまとめ

- 確率的潜在意味スケーリング(PLSS):
テキストを任意の潜在的な軸で連続的に測定する
 - 項目反応理論をベースにした、LSSの確率拡張
 - LSIからPLSI (Hofmann 1999)への拡張と同じ方向
- 極性辞書または少数の代表テキストから、測定軸を推定
- 公開されている尺度データでLSSより高性能、国会議事録など、様々な生テキストで実験
- 関連して、キーワードに関係する文書を文書ベクトルを用いて、統計的に選択するための正規-コーシー混合モデルを提案している (論文参照)

全体のまとめ

- 大規模言語モデルの背後にある、概念を表す単語ベクトルと状態ベクトルの幾何を説明した
 - 個々のテキストの内容はLLMで分析できるが、全体の統計的な傾向をとらえる・数学的な根拠のある判断を行うには、**統計的な分析**が必要
- LLMの内部に踏み込むには、“**数学的に手を入れる**”ことが不可欠
- LLMがもっと複雑なタスクを実行している際の回路の解析など、科学的にやることは山積み
- Python/Rの実装とJupyter Notebookは、岩波教科書のサポートサイトで公開しています