

2024年9月7日,にいがた"知の革新"STELLAプログラム
於 : zoom

AI活用入門

研究と社会のつながり

資料作成・講師 : 新潟大学BDA研究センター 齋藤裕

本日の講座の趣旨

✕ AIに関する知識を身に付ける

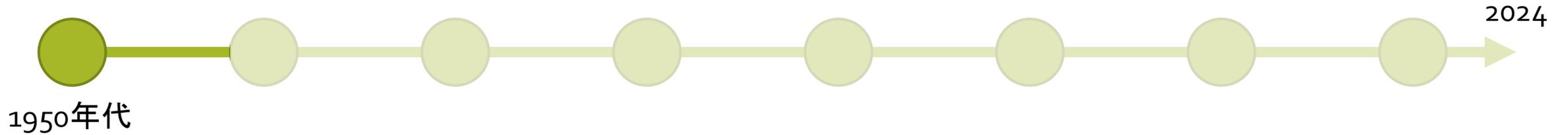
○ AI技術の発展と社会実装の流れから
研究と社会のつながりを学ぶ。

○ AI技術への関心を高め、積極的な研究活用を促す。

人工知能の歴史概要

1950年代	チューリングテストの提唱(<i>Alan Turing</i>) "Artificial Intelligence" という言葉が生まれる (<i>John McCarthy</i>)
1960年代	初期のAIチャットボット「ELIZA」 (第1次AIブーム)
1970年代	知識ベースシステム (knowledge base)
1980年代	エキスパートシステム (~2000頃、第2次AIブーム)
1990年代	機械学習理論の発展
2000年代	ビッグデータの利用、社会実装
2010年代	深層学習(Deep Learning)モデルのブレイクスルー、社会実装 言語モデルの発展
2020年代	生成モデルの発展、社会実装

人工知能(AI)という概念の誕生



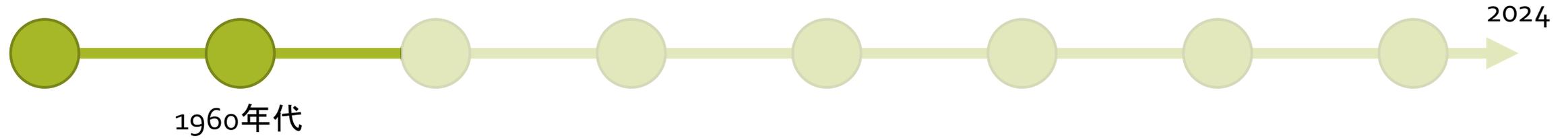
「Artificial Intelligence」

(John McCarthy, 1955)

“the science and engineering of making intelligent machines”

This image by: Wikimedia Commons, CC-BY-SA-2.0,
“https://commons.wikimedia.org/wiki/File:John_McCarthy_Stanford.jpg”

AIブーム①

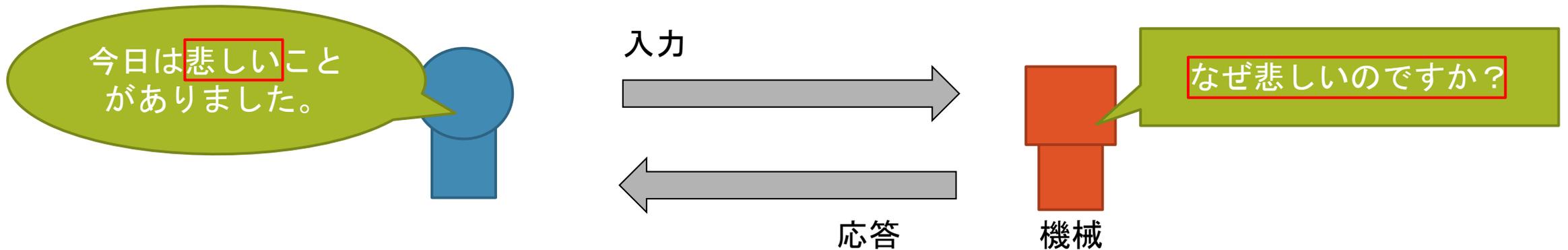


- 初期AI「ELIZA」 (*Joseph Weizenbaum, 1966*)

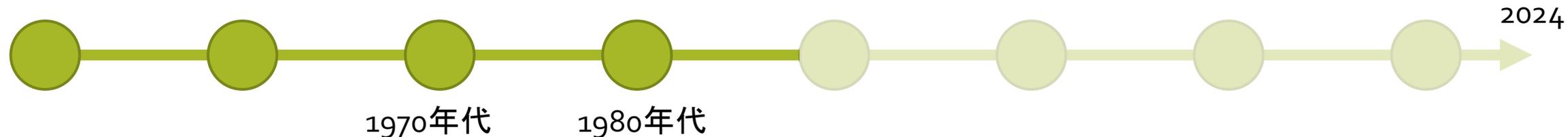
「推論」と「探索」による対話型AI.

ユーザーの入力に対して決まったパターンで応答することで対話を模倣。

(第1次)AI実装ブームを起こした社会実装の一形態。ただし決められた応答パターンしかもたない。

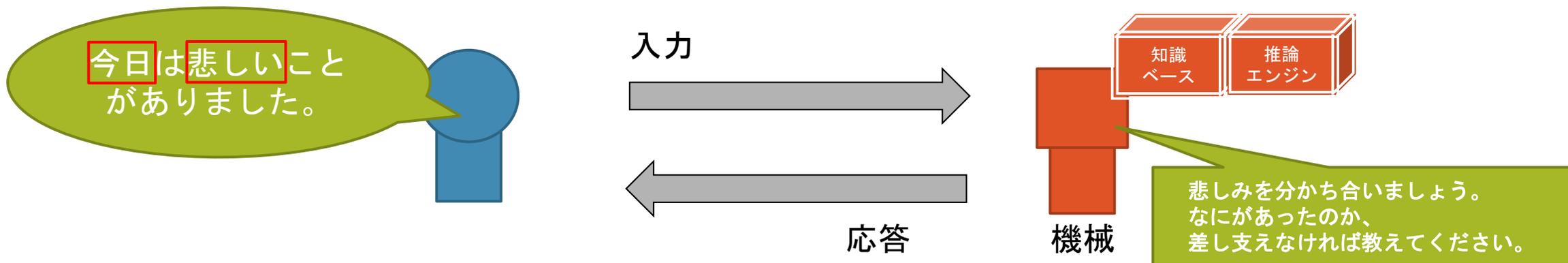


AIブーム②

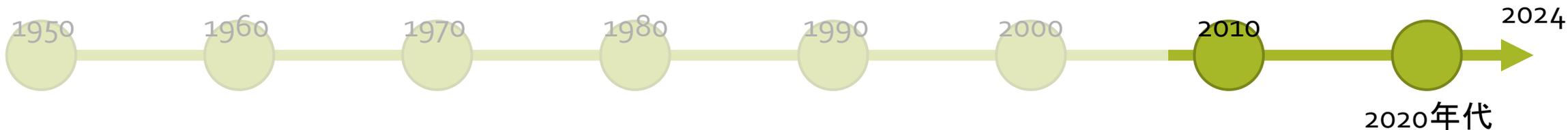


- エキスパートシステム

(第2次) **AI実装ブーム**を起こした社会実装の一形態。
知識ベースに高度な推論エンジンと対話型インターフェースをつけたもの。
比較的カンタンな操作で高度な知識を含む自然な対話ができる。ただし事前準備が大変。



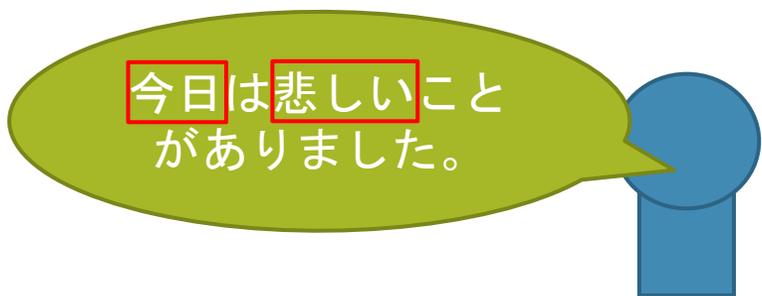
AIブーム③



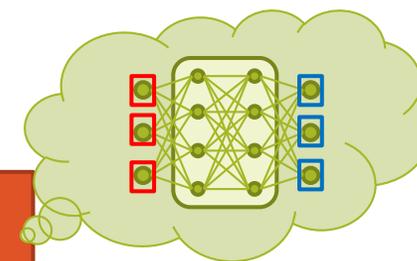
ChatGPT リリース (OpenAI社、2022.11)

- ✓ 準備の手間がいらぬ
- ✓ 誰でもすぐに使用できる
- ✓ 実用上の効果アリ

- + 後発のGoogleがリリースデモで躓いた
- + MicrosoftがWindowsに載せた



入力



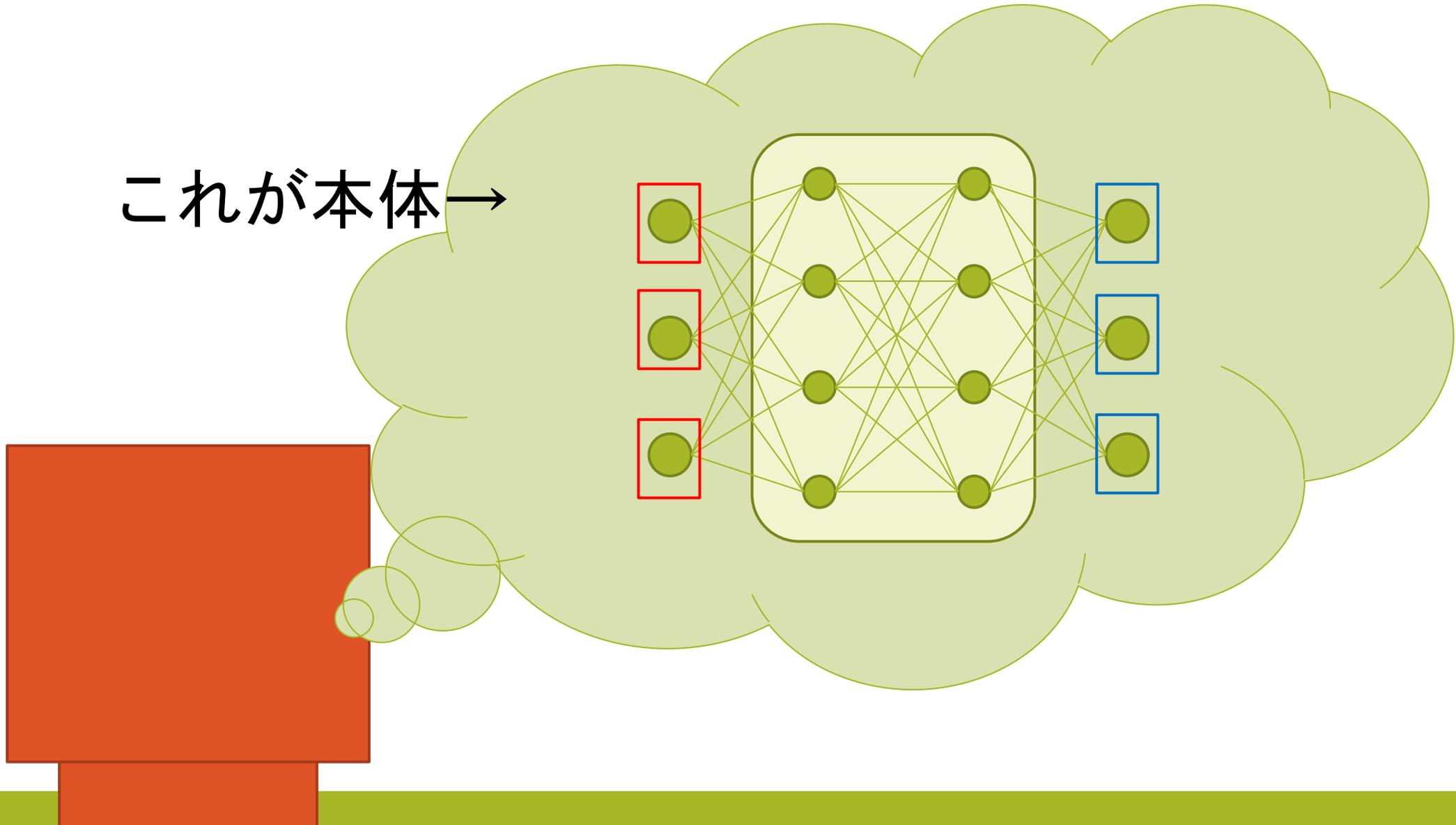
応答

悲(悲)しい(しい)こと(こと)があ(あ)った(っ)ん(ん)です(す)ね(ね)。そ(そ)れ(れ)は(は)大(大)変(変)で(で)した(し)た(た)ね(ね)。何(何)か(か)お(お)話(話)し(し)たい(たい)こ(こ)と(と)が(が)あ(あ)れ(れ)ば(ば)、ぜ(ぜ)ひ(ひ)聞(聞)か(か)せ(せ)て(て)く(く)だ(だ)さ(さ)い(い)。少(少)し(し)で(で)も(も)気(気)持(持)ち(ち)が(が)軽(軽)く(く)な(な)る(る)お(お)手(手)伝(伝)い(い)が(が)でき(でき)れ(れ)ば(ば)と(と)思(思)い(い)ま(ま)す(す)。

OpenAI, ChatGPT 4o(2024/06/07 14:00:00生成)

ただし性能が良すぎた。

これが本体→



本日のポイント

(研究) ChatGPTのリリースに至るまでの流れ

(実装) AIの実装に伴う社会的課題・事例

(活用) AI技術を「活用」する前に

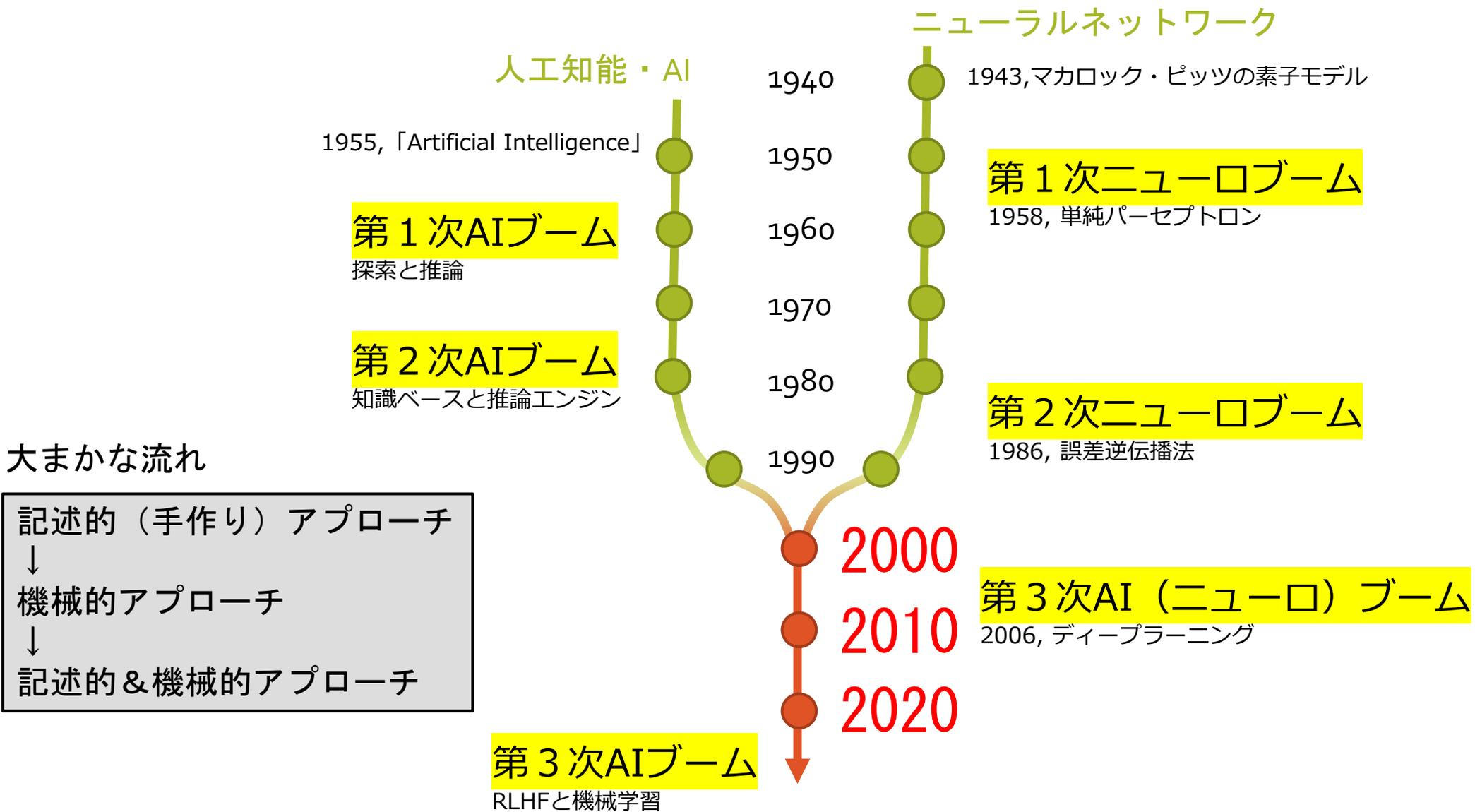
本日のポイント

観点

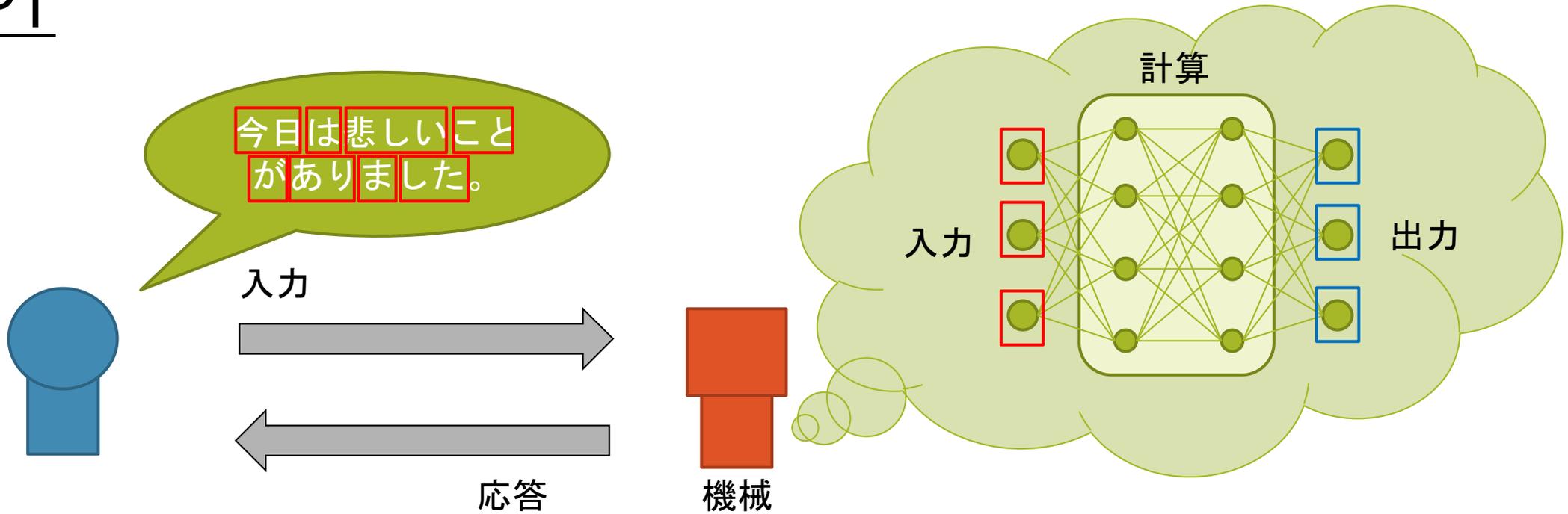
(**研究**) ChatGPTのリリースに至るまでの流れ

(**実装**) AIの実装に伴う社会的課題・事例

(**活用**) AI技術を「活用」する前に



ChatGPT

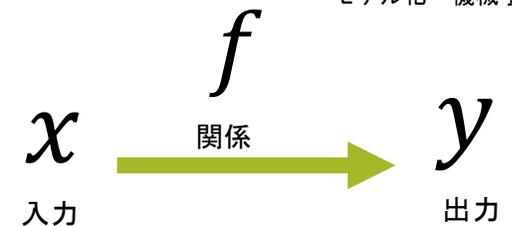


悲しいことがあったんですね。それは大変でしたね。何かお話ししたいことがあれば、ぜひ聞かせてください。少しでも気持ちが軽くなるお手伝いができると思います。

OpenAI, ChatGPT 4o(2024/06/07 14:00:00生成)

この入出力の関係を
どうやって作るか？

探索・推論・知識
エキスパートシステム
モデル化 機械学習



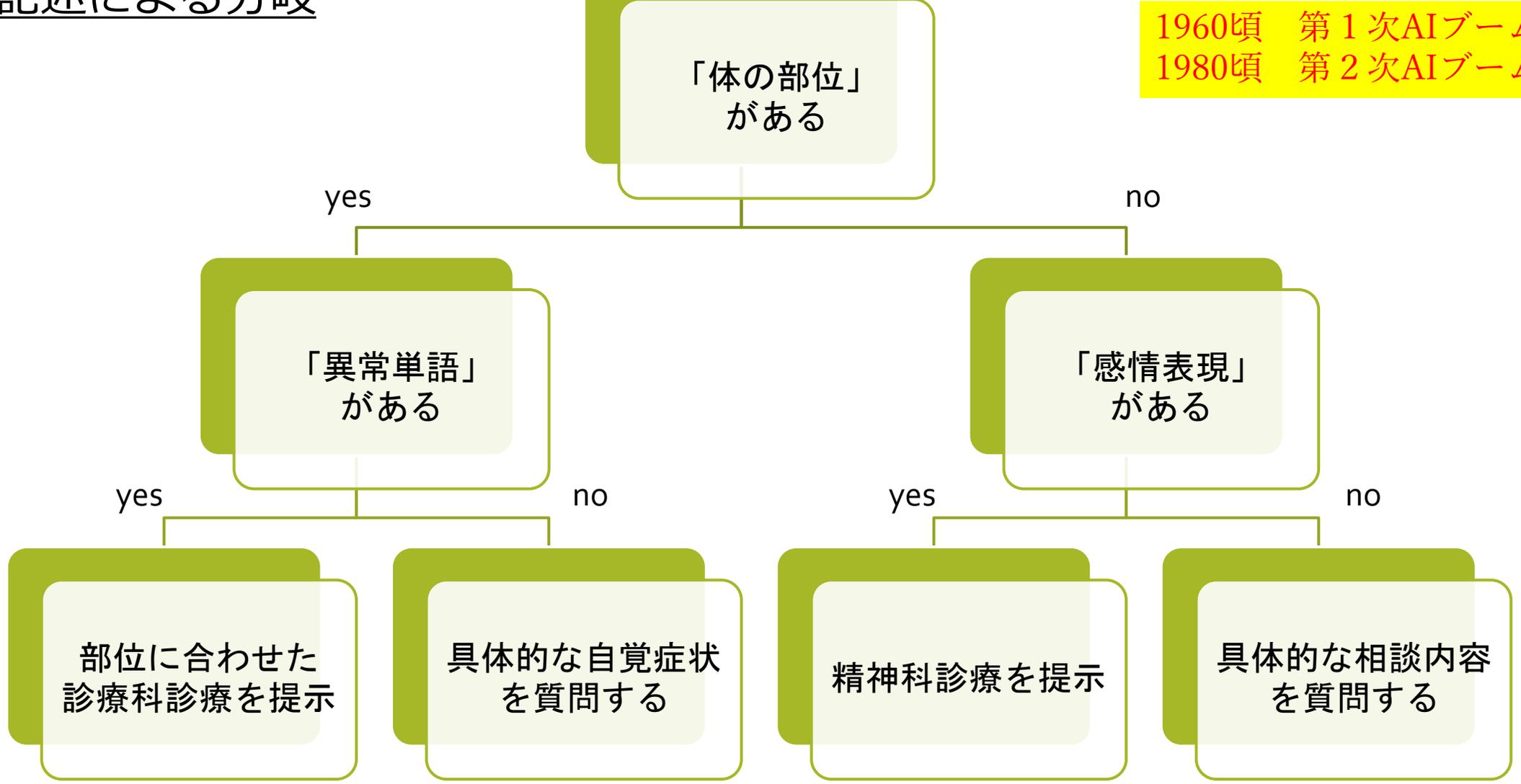
この入出力の関係を
どうやって作るか？



条件記述による分岐

入力

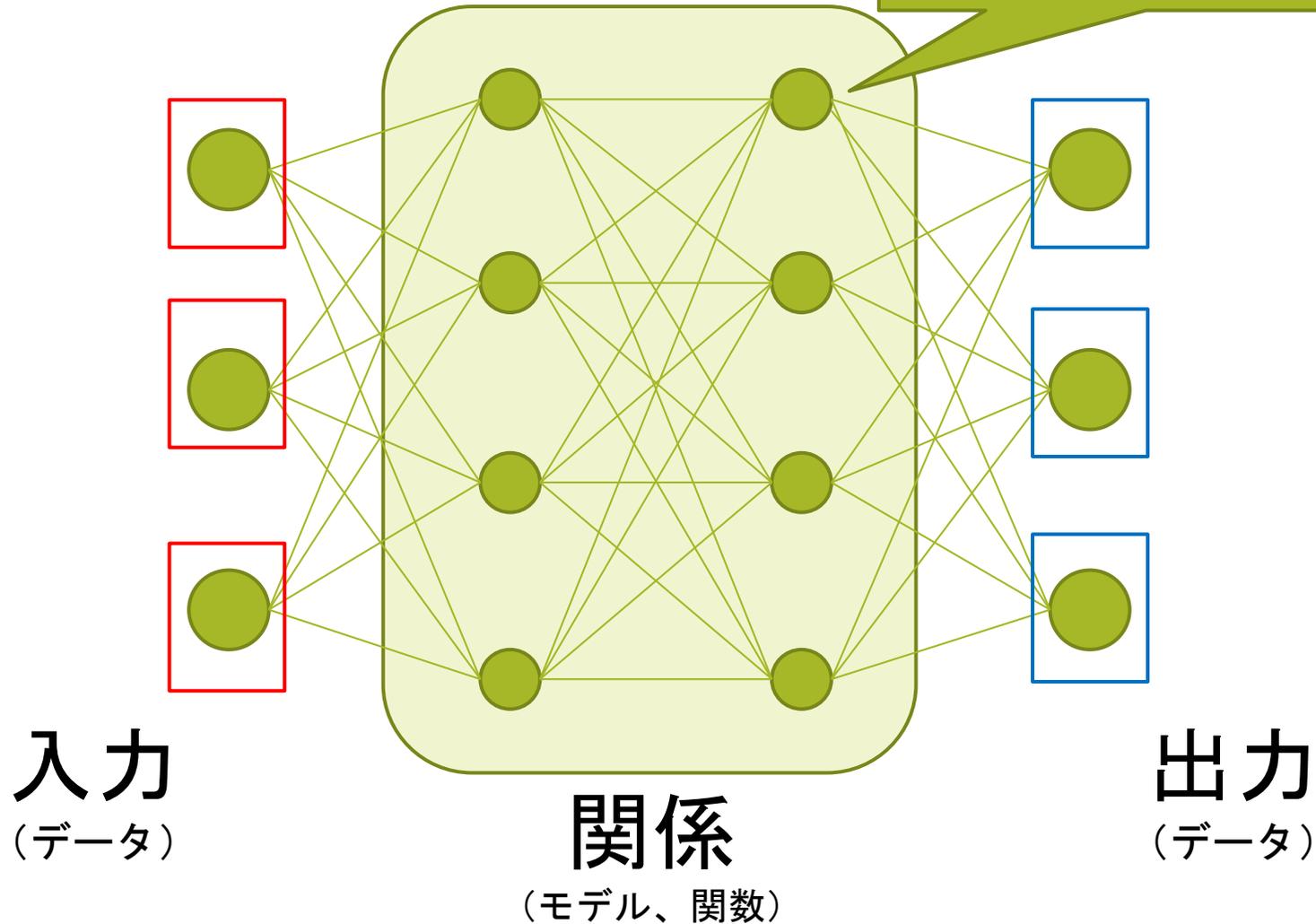
1960頃 第1次AIブーム
1980頃 第2次AIブーム



出力

多層ニューラルネットワーク (2006)

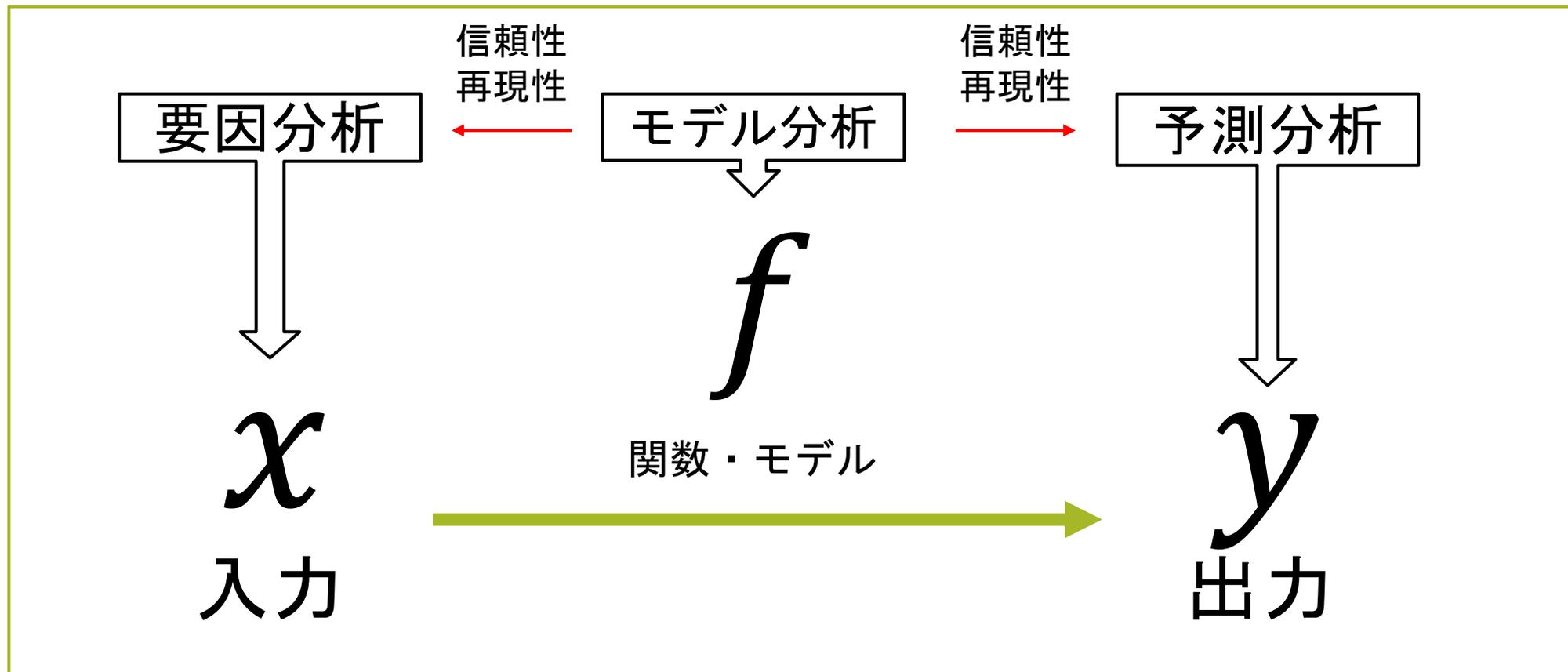
ここを複雑にすると
どんな"関係"でも
表現できる！！



ビッグ
データ

課題解決に必要な知見の抽出

2010頃 第3次AIブーム



過学習

データの
品質

計算コスト

ブロック
ボックス

学習バイアス

技術的課題

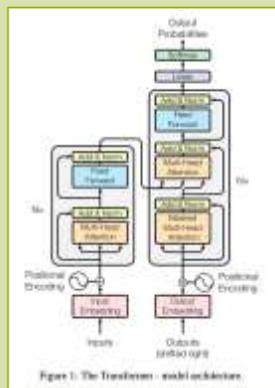
トランスフォーマー

Transformer and Attention

- ・ 計算時間を大幅短縮
- ・ 性能がよい
- ・ 意味解釈がしやすい (組み合わせ易い)

複雑なモデルを **簡単に** 作れる！

A. Vaswani et al., Attention is all you need,
<https://doi.org/10.48550/arXiv.1706.03762>, **2017**.



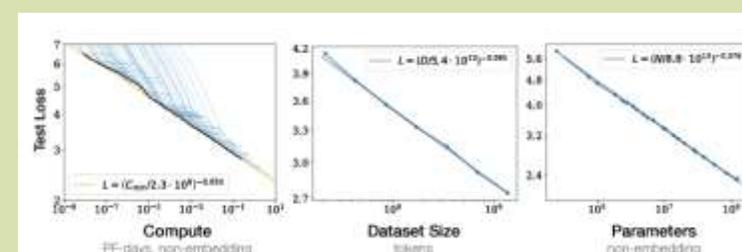
べき乗則

Scaling Law

- ・ データ量
 - ・ 学習量
 - ・ モデルの複雑さ
- の3つを増加させれば

性能が無限に向上する！

Model	Author	Year	Compute (PF-days)	Dataset Size (tokens)	Parameters
OpenAI GPT-1	OpenAI	2020	~3.6e12	~1.6e12	~1.5e11
OpenAI GPT-2	OpenAI	2019	~8.0e11	~1.5e12	~1.5e12
OpenAI GPT-3	OpenAI	2020	~3.6e12	~1.6e12	~1.5e12
OpenAI GPT-4	OpenAI	2023	~1.3e14	~1.6e12	~1.5e12



J. Kaplan et al., Scaling laws for neural language models, <https://doi.org/10.48550/arXiv.2001.08361>, **2020**.

GPT

Generative Pre-trained Transformer

Improving Language Understanding by Generative Pre-Training

Alec Radford
OpenAI
alec@openai.com

Karthik Narasimhan
OpenAI
karthikn@openai.com

Tim Salimans
OpenAI
tim@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

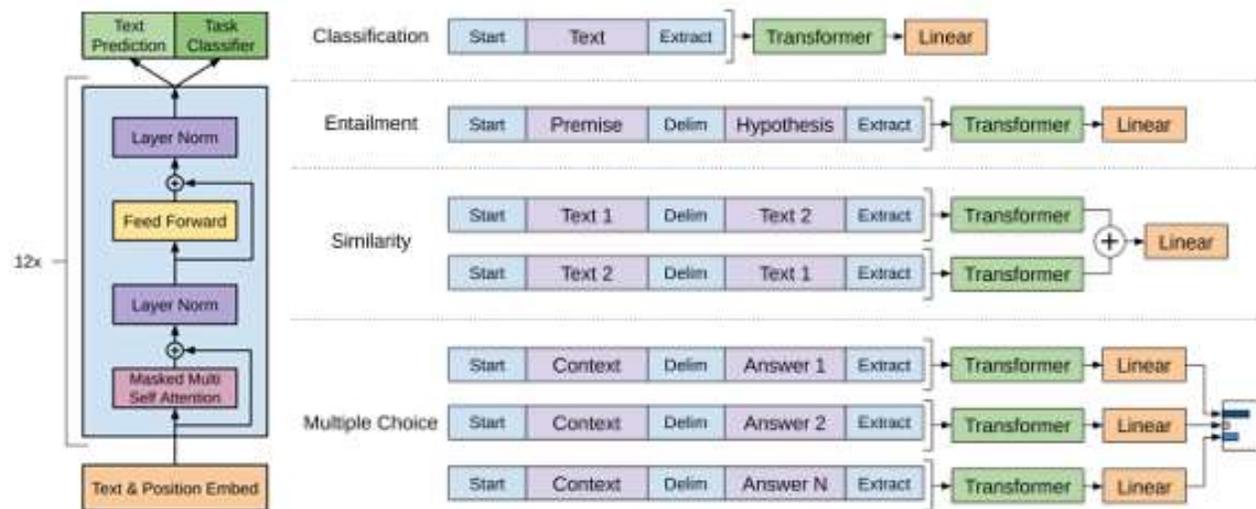


Figure 1: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Radford et al., Improving Language Understanding by Generative Pre-Training, **2018**.

GPTメジャーバージョンの歴史

パラメータ数
(モデルの複雑さ)

2018.06	GPT (アーキテクト) の論文発表 GPT-1 (モデル) リリース	1.17億
2019.02	GPT-2 プレリリース	15億
.11	(フルリリース)	
2020.05 .06	GPT-3 プレリリース (フルリリース)	1750億
2021	(InstructGPT)	
2022.05 .11	GPT-3.5 & API有償サービス リリース ChatGPT リリース	非公開
2023.03	GPT-4 リリース Microsoft Copilot リリース	非公開
2024.05	GPT-4o リリース	非公開

べき乗則
の発見

本日のポイント

(**研究**) ChatGPTのリリースに至るまでの流れ

観点

(**実装**) AIの実装に伴う社会的課題・事例

(**活用**) AI技術を「活用」する前に

➤ 第3次AIブーム（2010年頃～）

研究者や開発者界隈の「AIブーム」は2010年前後から。
ニューラルネットワーク技術のブーム。

=第3次ニューロブーム

社会的な「AIブーム」は2023年～

ChatGPTのリリースからなる生成AI技術によるブーム。

製品開発プロセス

1. 企画立案
2. 市場調査・販売戦略策定
3. 技術テスト（プロトタイプ）
4. 試用テスト・製品ブラッシュアップ
5. 商品化・販売

➤ Amazon 人事採用アルゴリズムの性差別 (2014～)

過去の履歴書と採用データに基づくアルゴリズムを人事採用システムとして使用していた。

2018年、女性差別的傾向があったとして使用を停止した。

母集団が男性偏重であったことによる

データバイアスが原因であったとされる。

➤ Google 「Google Photos」 (2015.5.29~)

写真に自動でタグをつけて管理する機能をもったアプリ。

ある黒人ユーザーが自撮り写真に「Gorilla」とタグをつけら

れたと発信し、機械学習と人種差別をめぐる議論となった。

現在でもGoogle Photosはゴリラの画像に「Gorilla」ラベルが付きにくいといわれる。

- Microsoft 「Tay」 (2016.3.23～同.3.25, 同3.30再開同日停止)

Twitter (現X) 上の書き込みBot。

Twitterでの対話から学習する能力を持った人工知能であったが、**悪意あるユーザーの影響により社会的に不適切な書き込み**をするようになり、公開が停止された。

- 日本マイクロソフト→株式会社りんな 「りんな」 (2015～)

LINE上のチャットbot。

2018年から「共感モデル」により文章生成を取り入れた。

過去のAI実装事例から不適切な学習は開発陣によって排除またはルールを学習させる処置がとられた。

利用リテラシー

フレーム
問題

倫理・法

説明可能性
(explainability)

説明責任
(accountability)

社会的課題

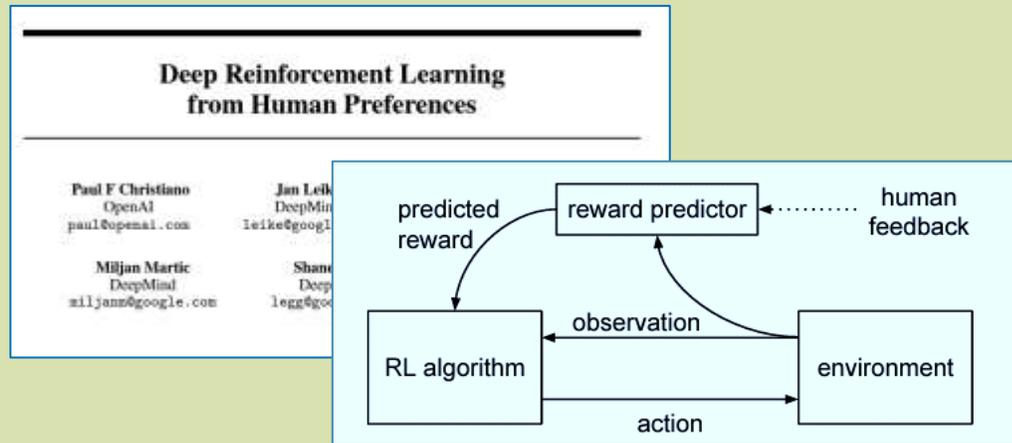
人間によるフィードバック (RLHF)

Reinforcement Learning from Human Feedback

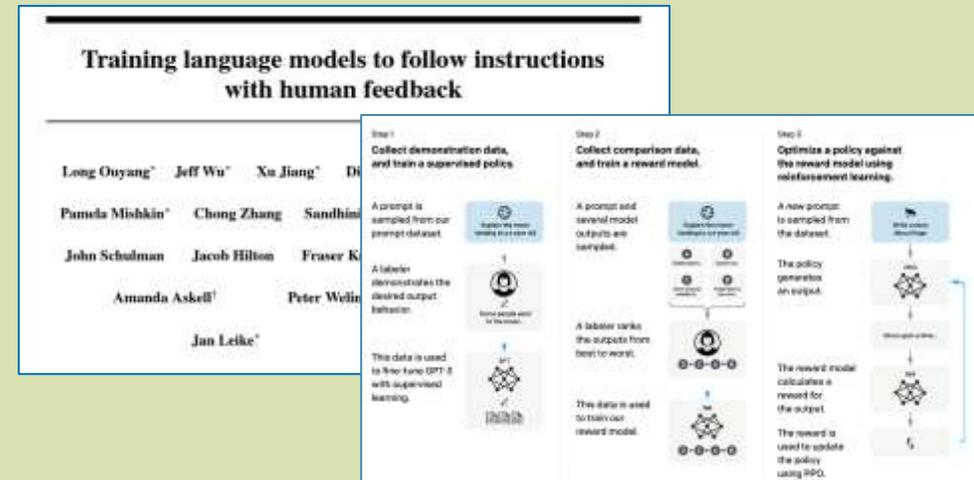
判断に困ったら人間に聞けばいい！



事前に人間の判断を教えておこう！



P.F.Christiano et al., Deep Reinforcement Learning from Human Preference, <https://doi.org/10.48550/arXiv.1706.03741>, **2017**.



L. Ouyang et al., Training language models to follow instructions with human feedback, <https://doi.org/10.48550/arXiv.2203.02155>, **2022**.

- データの前処理

データ収集、クレンジング、
特徴量エンジニアリングなど

- モデル設計

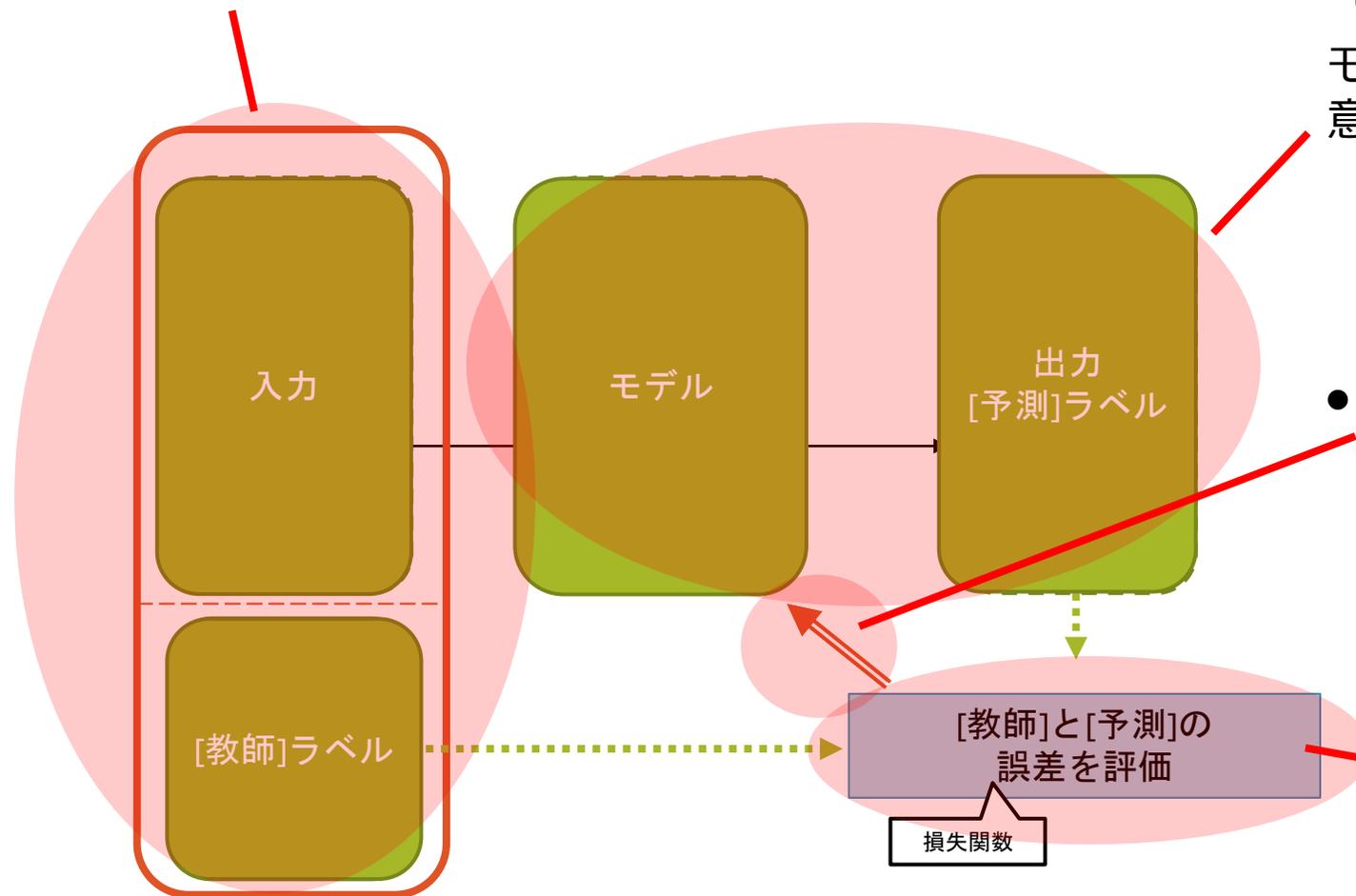
モデルの選択、構造設計、
意味空間設計など

- パラメータ調整

反復回数や学習率の調整、
パラメータサーチ、ログ管理など

- 評価の定義

損失関数の選定、意味解釈など



GPTメジャーバージョンの歴史

パラメータ数
(モデルの複雑さ)

2018.06	GPT (アーキテクト) の論文発表 GPT-1 (モデル) リリース	1.2億
2019.02	GPT-2 プレリリース	15億
.11	(フルリリース)	
2020.05	GPT-3 プレリリース	1750億
.06	(フルリリース)	
2021	(InstructGPT)	
2022.05	GPT-3.5 & API有償サービス リリース	非公開
.11	ChatGPT リリース	
2023.03	GPT-4 リリース Microsoft Copilot リリース	非公開
2024.05	GPT-4o リリース	非公開

べき乗則
の発見

RLHF
導入

2023.09
Red Teaming
Network 設置

本日のポイント

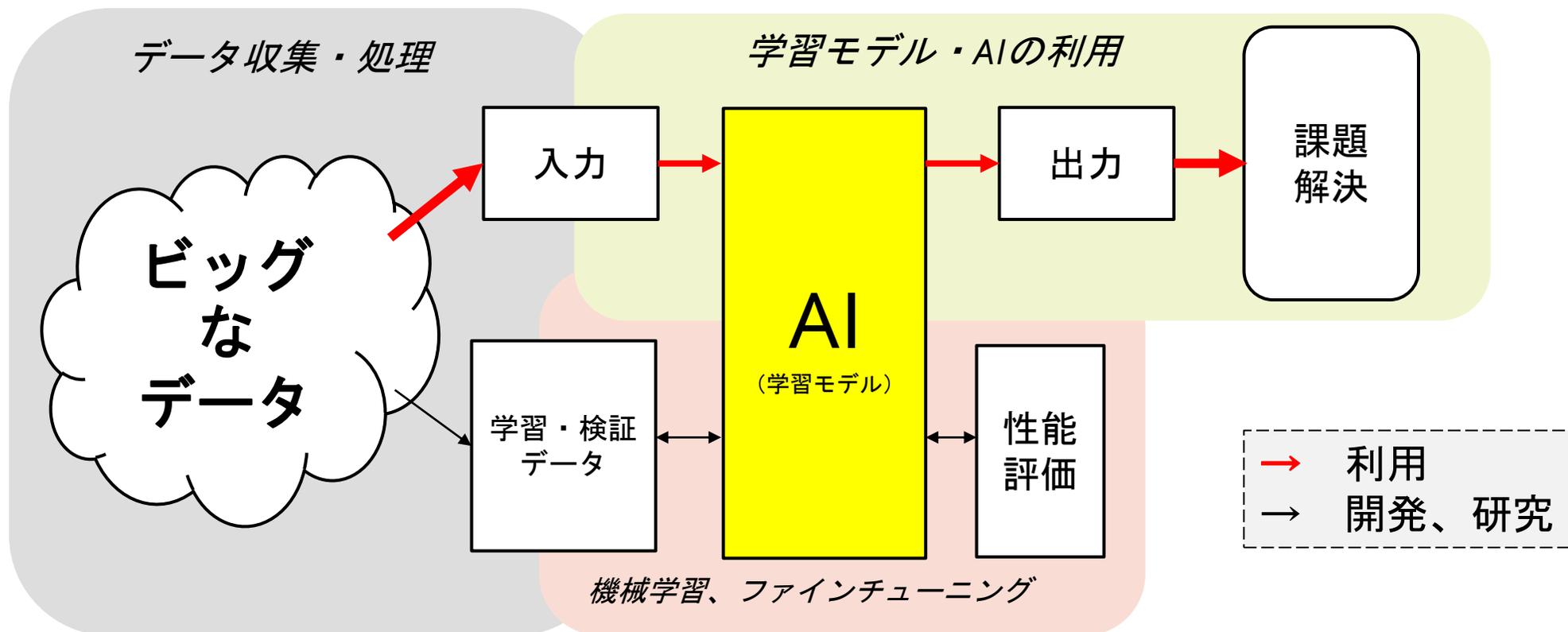
(**研究**) ChatGPTのリリースに至るまでの流れ

(**実装**) AIの実装に伴う社会的課題・事例

観点

(**活用**) AI技術を「活用」する前に

「使う」と「作る」の違い



モデル

学習データ

学習モデル

学習パラメータチューニング

追加学習

API/Local

システム

通信制御

計算パラメータチューニング

入力制御

出力制御

SaaS
webブラウザ

ユーザ

入力データ

表示閲覧

実質制御不可

- ・ 学習データの取捨選択
- ・ 学習プロセスの透明化
- ・ ELSI諸問題

システム・規約・ガイドラインで制御

- ・ プロンプト/計算パラメータ調整
- ・ 個人情報仮名化などのデータ処理
- ・ 利用ログの管理
- ・ 不適切な出力のフィルタ

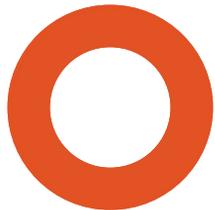
- ・ 情報リテラシーの修得
- ・ データリテラシーの修得
- ・ AIリテラシーの修得

(テキスト生成AI利用レベル) AIを使う前に目的とタスクを設定



ダメな例：

- 翻訳をする
- 文章を書く
- 分類する



タスクの例：

- 日本語で書かれた200字ほどの業務依頼のメール文章を英語に翻訳をする
- フライヤーに掲載するセミナーの概要として100程度の文章を書く
- メディア別に収集したwebテキストの想定読者年齢を推定して分類する

過学習

学習済
モデルの
権利

データ
の品質

計算コスト

データの
権利

ブラック
ボックス

学習バイアス

計算資源
の確保

ハルシネーション
(事実と錯覚する虚偽)

犯罪利用

著作権侵害

情報流出
(フロントインジェクション)

汎用人工知能

**労働需要
の喪失**

**フェイクの
乱造製造**

データ汚染

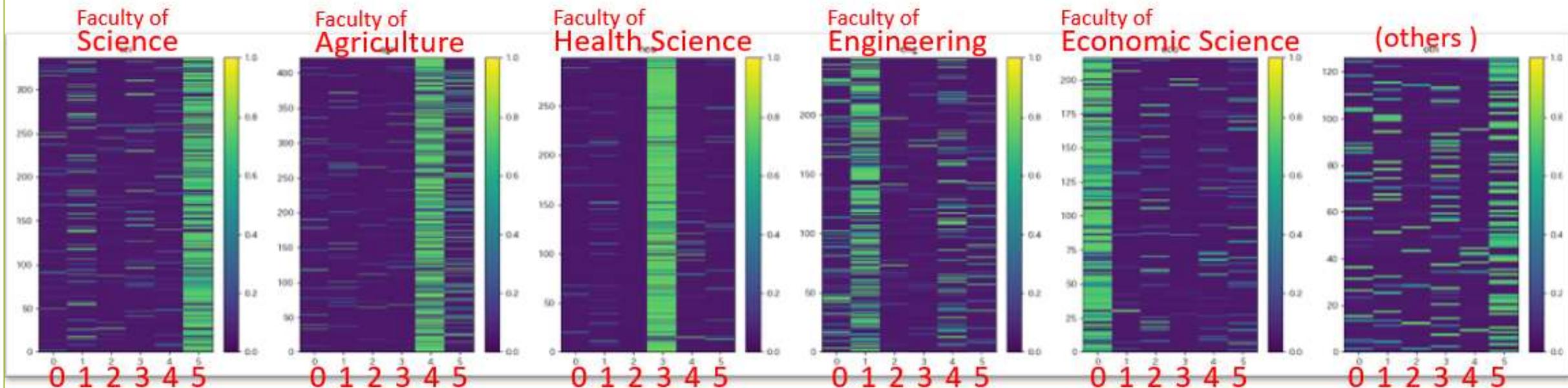
環境問題

(研究紹介)

教育 × **AI** × **データサイエンス**

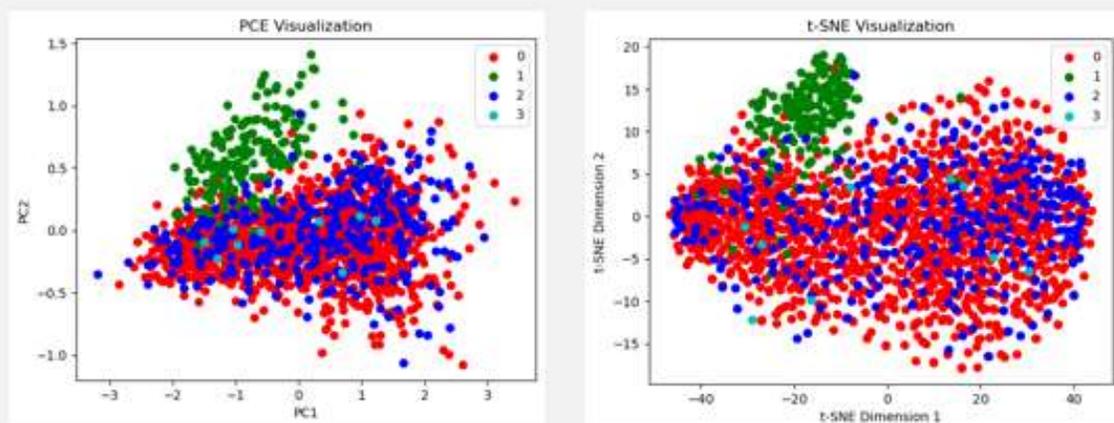
Topic classification by LDA (Answers_2)

topic #0: **economics, company, expenditure, ...** 経済, 企業, 消費, 経営, 分析, 統計, 売り上げ, 都市, 行動, 建築
 topic #1: **engineering, learning, intelligence, ...** 工学, 学習, 知能, 大学, 画像, 処理, 人工, 技術, 機械, ゲノム-Genom
 topic #2: **merchandise, client, occur, ...** 商品, 顧客, 発生, 効果, 点, 分かる, 高い, 分析, 時間, 売り上げ
 topic #3: **patient, treatment, person, ...** 患者, 治療, 個人, 検査, カルテ-Karte, 病気, 電子, 法, 健康, 看護
 topic #4: **crops, production, management, Experiment, Science, Physics, ...** 作物, 生産, 管理, 栽培, 作業, AI, 量, 技術, 自動, 収穫
 topic #5: 実験, 理学, 物理, 化学, 部, 数学, 生物, 統計, 調査, サイエンス-science

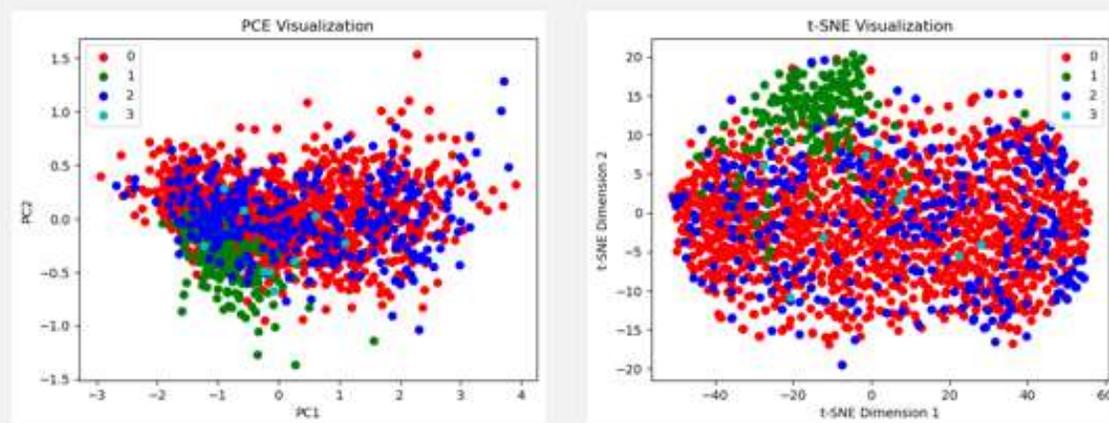


PCE and t-SNE of latent space by word2vec

Ans_1



Ans_2



- AI
- Student(training data)
- Student(evaluation data)
- Student(AI used, N=9)

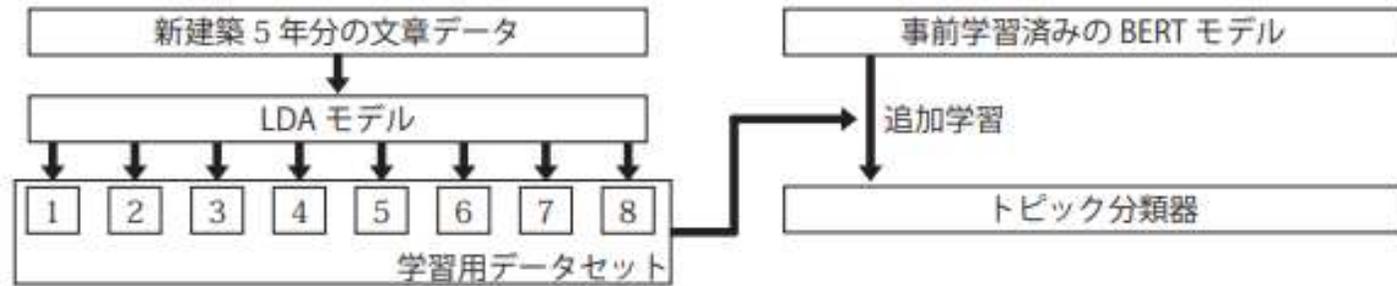


fig. 1 トピック分類器の作成

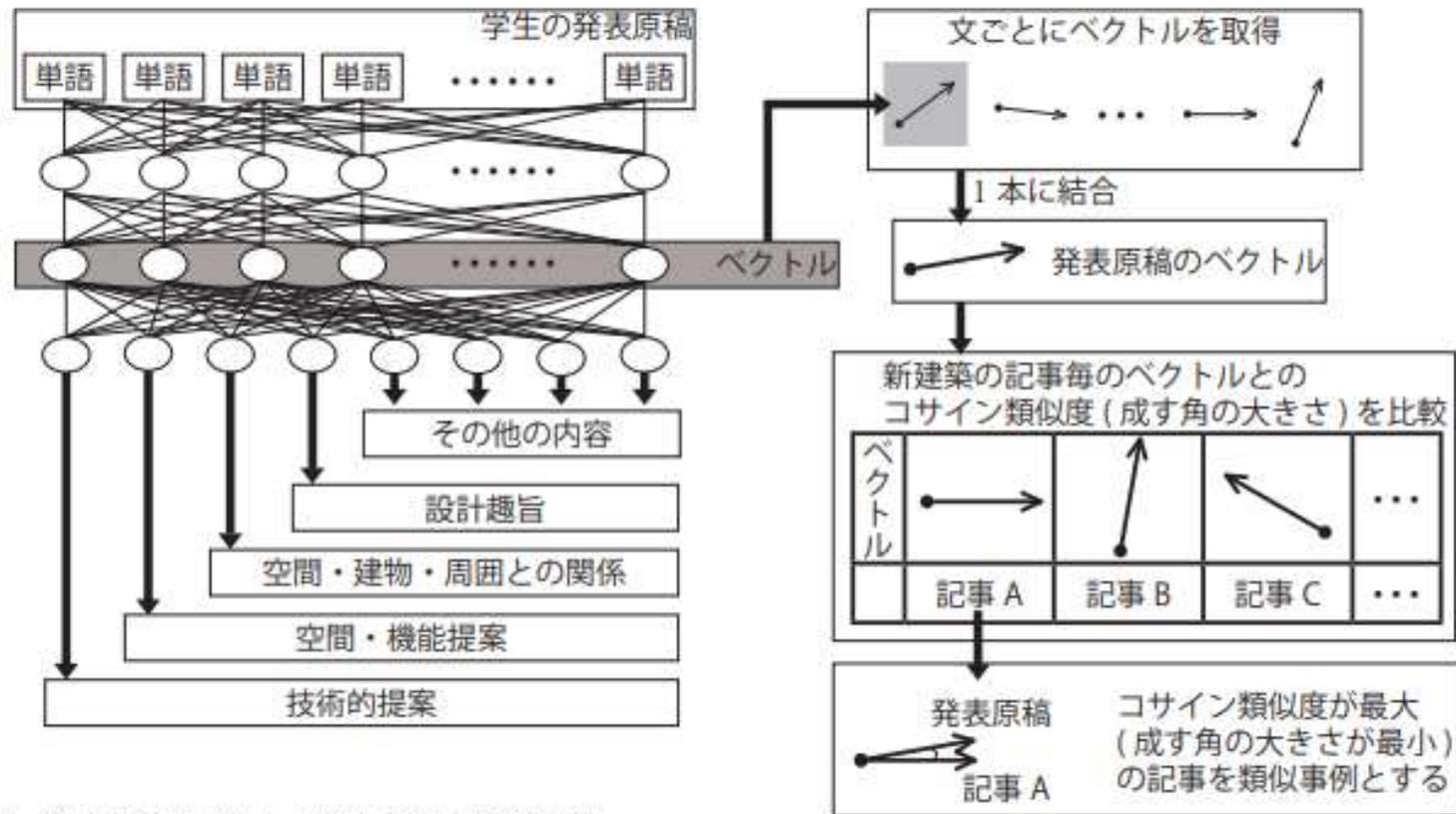


fig. 2 発表原稿のベクトル化と類似事例の選定

(エラー文から始まるプログラミング学習イメージ)

入力

```
Cell In[22], line 1  
print("Hello World!")  
      ^
```

SyntaxError: incomplete input

次にpythonのエラー文を提示するので、エラーの理由を説明し、解決方法を提示してください。¥n 解答形式===¥n エラーの理由：¥n 解決方法： ¥n===

プロンプト制御

エラーの理由：

このエラーメッセージは、Pythonで構文エラー（SyntaxError）が発生していることを示しています。具体的には、文字列が正しく閉じられていないためです。Pythonでは、文字列はシングルクォート（'）またはダブルクォート（"）で始まり、同じ記号で終わる必要があります。この場合、ダブルクォートで始まった文字列がダブルクォートで閉じられていないためエラーが発生しています。

解決方法：

~~この問題を解決するには、文字列を適切に閉じる必要があります。つまり、不足しているダブルクォート（"）を行末に追加することで、文字列を正しく閉じます。~~

このエラーメッセージは、Pythonで構文エラー（SyntaxError）が発生していることを示しています。具体的には、文字列が正しく閉じられていないためです。Pythonでは、文字列はシングルクォート（'）またはダブルクォート（"）で始まり、同じ記号で終わる必要があります。この場合、ダブルクォートで始まった文字列がダブルクォートで閉じられていないためエラーが発生しています。

理由だけ
指導して

解決方法は不表示

システム制御



本日のまとめ

- AI技術の発展と社会実装の流れから研究と社会のつながりを学ぶ。
- AI技術への関心を高め、積極的な研究活用を促す。

2024年9月7日,にいがた"知の革新"STELLAプログラム
於 : zoom

AI活用入門

研究と社会のつながり

資料作成・講師 : 新潟大学BDA研究センター 齋藤裕