

# ae-11.人工知能と自然言語処理 の基礎

(AI 演習) (全15回)

<https://www.kkaneko.jp/ai/ae/index.html>

金子邦彦



# 人工知能（AI）の学習のための指針



1. **実践重視**： AIツールを実際に使用し、機能に慣れる
2. **エラーを恐れない**： 実行においては、エラーの発生の可能性はある。エラーを恐れず、むしろ学習の一部として捉えるポジティブさが大切。
3. **段階的学習**： 基礎から応用へと段階的に学習を進め、AIの可能性を前向きに捉える



# アウトライン

1. イントロダクション
2. 自然言語処理と人工知能
3. 単語ベクトル
4. チャットボットの概要
5. チャットボットのベストプラクティス

# 11-1. イントロダクション

# 人工知能

知的なITシステム

## 機械学習

データから**学習**し、知的能力を  
向上

## ディープラーニング

データから**学習**し、複雑なタスク  
を実行。**多層のニューラル  
ネットワーク**を使用

# 機械学習の基本



**機械学習**は、**コンピュータ**が**データ**を使用して**学習**することにより**知的能力を向上**させる技術


- **情報の抽出**：データからパターンや関係性を自動で見つけ出す能力を持つ
- **知的なタスクの実行**：予測，分類などの知的なタスクを実行



# 機械学習の3つの特徴



1. **情報の抽出**：データからパターンや関係性を自動で見つけ出す能力を持つ
2. **簡潔さ**：人間が設定していたルール等を，自動生成できる
3. **限界の超越**：従来の方法では困難だった課題も解決できる可能性がある



## 11-2. 自然言語処理と 人工知能



# 自然言語処理と人工知能

**自然言語処理**は、人間が普段使う言語（日本語、英語など）を**コンピュータが理解**したり、**生成**したりする技術

- AI による**人間の言語の理解**
- AI と人間の**自然な対話**
- **言語間の翻訳**
- **文書類**（人間の言葉によるデータ）の**分析と活用**

# 自然言語処理の応用例

**情報検索**：インターネットからの情報検索（検索エンジンの活用）

**AIとの対話**：AIアシスタントとの対話や指示

**プログラミング支援**：コード作成，デバッグ，最適化

**文書作成・推敲**：文章の作成と改善

**翻訳**：言語間の翻訳（例：動画の自動翻訳，字幕付け）

**要約**：長文からのポイント抽出

# 翻訳（日本語から英語，英語から日本語）

The screenshot shows the DeepL translator interface. At the top, there are two buttons: "テキスト" (Text) and "ドキュメント" (Document). Below these are language selection menus. The source language is set to "日本語 - 自動検出" (Japanese - Auto-detect) and the target language is "英語" (English). The input text is "白い雲と青い空が美しい" (Shiroi kumo to aoi sora ga utsukushi). The output text is "Beautiful white clouds and blue sky". There are icons for voice input/output, a character count of 11/5000, and a feedback button.

日本語 - 自動検出 英語 日本語 韓国語 ▼ ↔ 日本語 英語 韓国語 ▼

白い雲と青い空が美しい × Beautiful white clouds and blue sky ☆

Shiroi kumo to aoi sora ga utsukushi

11/5000

フィードバックを送信

The screenshot shows the DeepL translator interface. At the top, there are two buttons: "テキスト" (Text) and "ドキュメント" (Document). Below these are language selection menus. The source language is "英語" (English) and the target language is "日本語" (Japanese). The input text is "Once upon a midnight dreary, while I pondered, weak and weary, Over many a quaint and curious volume of forgotten lore, |". The output text is "深夜の惨めな時に、私が熟考しながら、弱くて疲れきった、忘れられていた伝説の古風で不思議なボリュームの上に、Shin'ya no mijimena toki ni, watashi ga jukukō shinagara, yowakute tsukare kitta, wasure rarete ita densetsu no kofūde fushigina boryūmu no ue ni,". There are icons for voice input/output, a character count of 120/5000, and a feedback button.

言語を検出する 英語 日本語 韓国語 ▼ ↔ 日本語 英語 韓国語 ▼

Once upon a midnight dreary, while I pondered, weak and weary, × 深夜の惨めな時に、私が熟考しながら、弱くて疲れきった、 ☆

Over many a quaint and curious volume of forgotten lore, | 忘れられていた伝説の古風で不思議なボリュームの上に、

Shin'ya no mijimena toki ni, watashi ga jukukō shinagara, yowakute tsukare kitta, wasure rarete ita densetsu no kofūde fushigina boryūmu no ue ni,

# 翻訳（ファイルの使用）

ワード、パワーポイントファイルの中から、  
文字のデータを取り出すことも当たり前

## ue-2. C++ プロジェクトの新規作成 (Unreal Engine 演習シリーズ)

<https://www.kkaneko.jp/cc/ue/index.html>

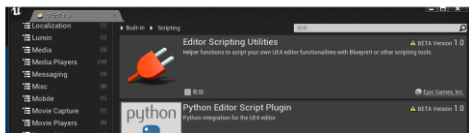
金子邦彦



### ① Unreal エディタのメインメニューで、「編集」、「プラグイン」と操作



### ② 「Scripting」を選び、「Python Editor Script Plugin」の「有効」をチェック



## ue-2. Create a new C ++ project (Unreal Engine exercise series)

Kunihiko Kaneko

<https://www.kkaneko.jp/cc/ue/index.html>

1 In the main menu of the Unreal Editor, operate "Edit" and "Plug-in"

2 Select "Scripting" and check "Enable" in "Python Editor Script Plugin"

3 Confirm the message and "Yes"

- Create Project Visual Studio code is generated
- Visual Studio project is open

翻訳結果

元のパワーポイントファイル

# 自然言語処理の主な用途

## 1. 言語の効果的な処理

例：Google翻訳, DeepL翻訳による言語間の翻訳

## 2. 人間との自然な対話

例：SiriやAlexaによるユーザーとの対話

これらはすでに日常生活で広く利用されている。

# 情報活用スキル

## 情報活用の主要なスキル

- **情報収集の効率化**：インターネットからの効率的な情報獲得
- **リサーチ・スキル**：信頼できる情報源の識別
- **根拠に基づく思考**：情報の信頼性と偏りの確認習慣

# 情報エンジンの活用

検索エンジンでは

- キーワードや文章による検索が可能
- 関連情報の自動推薦, 広告表示
- AIによる概要の表示

例: 「サステナビリティ」の Google 検索エンジンによる検索

Google サステナビリティ

約 131,000,000 件 (0.30 秒)

サステナビリティ (sustainability) とは、環境や経済等に配慮した活動を行うことで、社会全体を長期的に持続させていこうという考え方です。「Sustain (維持する、持続する)」と「Ability (～する能力)」を組み合わせた造語で、日本語では「持続可能性」と呼ばれます。 2023/11/14

三井住友銀行  
https://www.smbc.co.jp/planning/about-sustainability

サステナビリティとは? 企業が取り組むメリットや指標

サステナビリティ 3つの柱

- 環境保護 (Environmental Protection)
- 社会開発 (Social Development)
- 経済発展 (Economic Growth)

持続可能性

持続可能性は、将来にわたって現在の社会の機能を継続していくことができるシステムやプロセスのこと。一般的にはそういった仕組みを指すが、環境学的には生物学的システムがその多様性と生産性を期限なく継続できる能力のことを指し、さらに組織原理としては、持続可能な発展を意味する。 ウィキペディア

他の人はこちらも検索 他 5 件以上を表示

# 情報活用における検索エンジンの利用

- 検索エンジンによる効果的な情報検索
- 演習を通じた検索スキルの向上
- 検索結果の信頼性と偏りの確認習慣





# 演習 1. 情報エンジンの 利用

# 情報検索の具体的な手順

## 1. トピックとキーワードの設定

- トピック例：脱炭素
- キーワード例：再生可能エネルギー，温室効果

## 2. 検索エンジンの活用

- Google検索の利用

<https://www.google.com/?hl=ja>

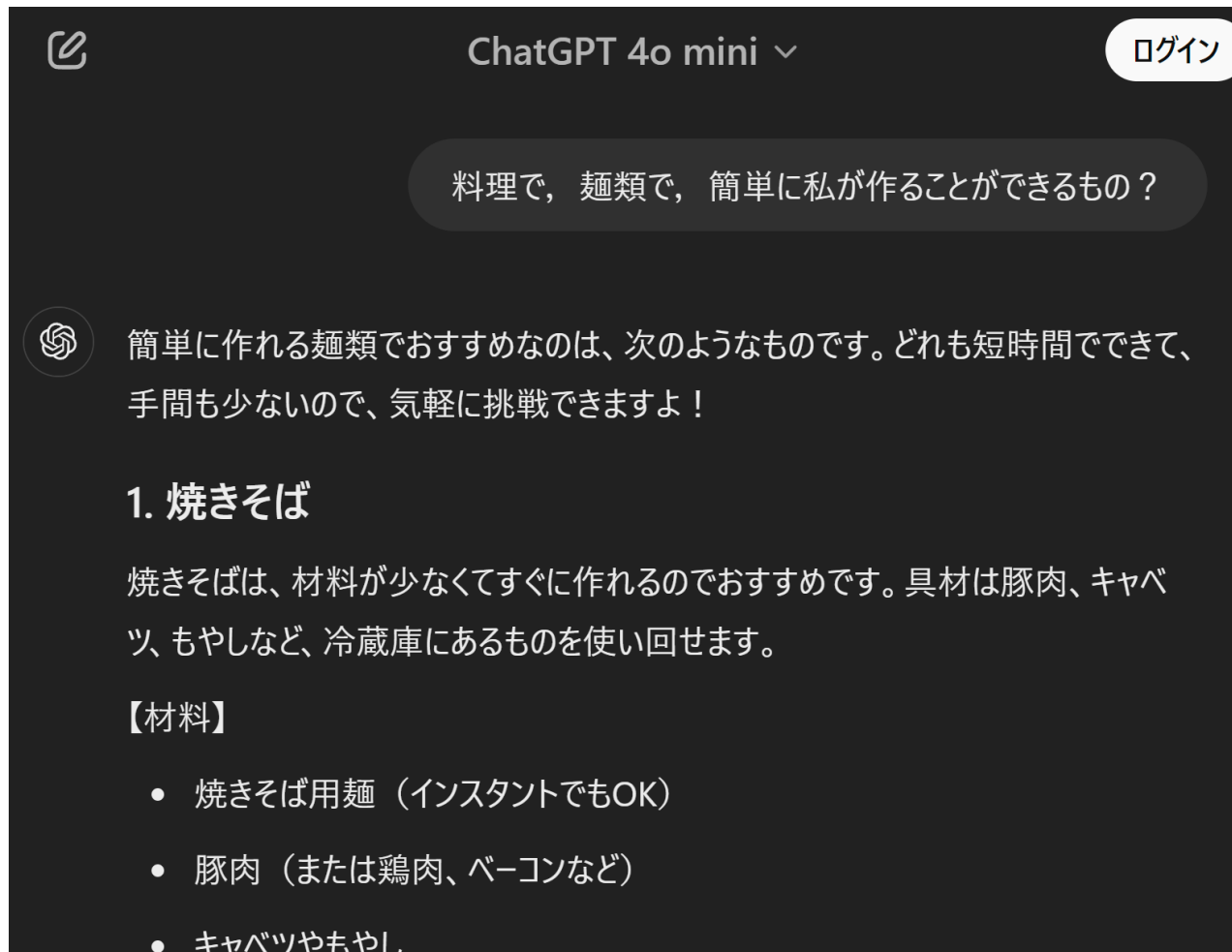
- （その他の検索園児に，他の情報源も必要に応じて併用）

## 3. 情報の評価

- 検索結果の信頼性と偏りの確認
- 当初の目的に沿うものか？ 新たな発見があるか？

# AI との対話

人間の言葉を使用したコミュニケーション, 自然な対話, 継続的な対話の維持, 追加質問可能, 複数のトピックへの対応



The screenshot shows a chat window for ChatGPT 4o mini. The user's prompt is: "料理で、麺類で、簡単に私が作ることができるもの？". The AI's response is: "簡単に作れる麺類でおすすめなのは、次のようなものです。どれも短時間でできて、手間も少ないので、気軽に挑戦できますよ！". The response is followed by a section titled "1. 焼きそば" and a paragraph: "焼きそばは、材料が少なくてすぐに作れるのでおすすめです。具材は豚肉、キャベツ、もやしなど、冷蔵庫にあるものを使い回せます。". Below this is a "【材料】" section with a list of ingredients: "焼きそば用麺 (インスタントでもOK)", "豚肉 (または鶏肉、ベーコンなど)", and "キャベツやもやし".

ChatGPT 4o mini ログイン

料理で、麺類で、簡単に私が作ることができるもの？

簡単に作れる麺類でおすすめなのは、次のようなものです。どれも短時間でできて、手間も少ないので、気軽に挑戦できますよ！

### 1. 焼きそば

焼きそばは、材料が少なくてすぐに作れるのでおすすめです。具材は豚肉、キャベツ、もやしなど、冷蔵庫にあるものを使い回せます。

【材料】

- 焼きそば用麺 (インスタントでもOK)
- 豚肉 (または鶏肉、ベーコンなど)
- キャベツやもやし

人間の  
プロンプト

AI の回答

# AIコミュニケーションスキル

AIとの効果的なコミュニケーションに必要なスキル

- プロンプトの工夫：効果的な質問・指示の方法
- AI技術の理解：AIの可能性と限界の把握
- 課題解決能力：AIを活用した問題解決

# AI アシスタントの活用



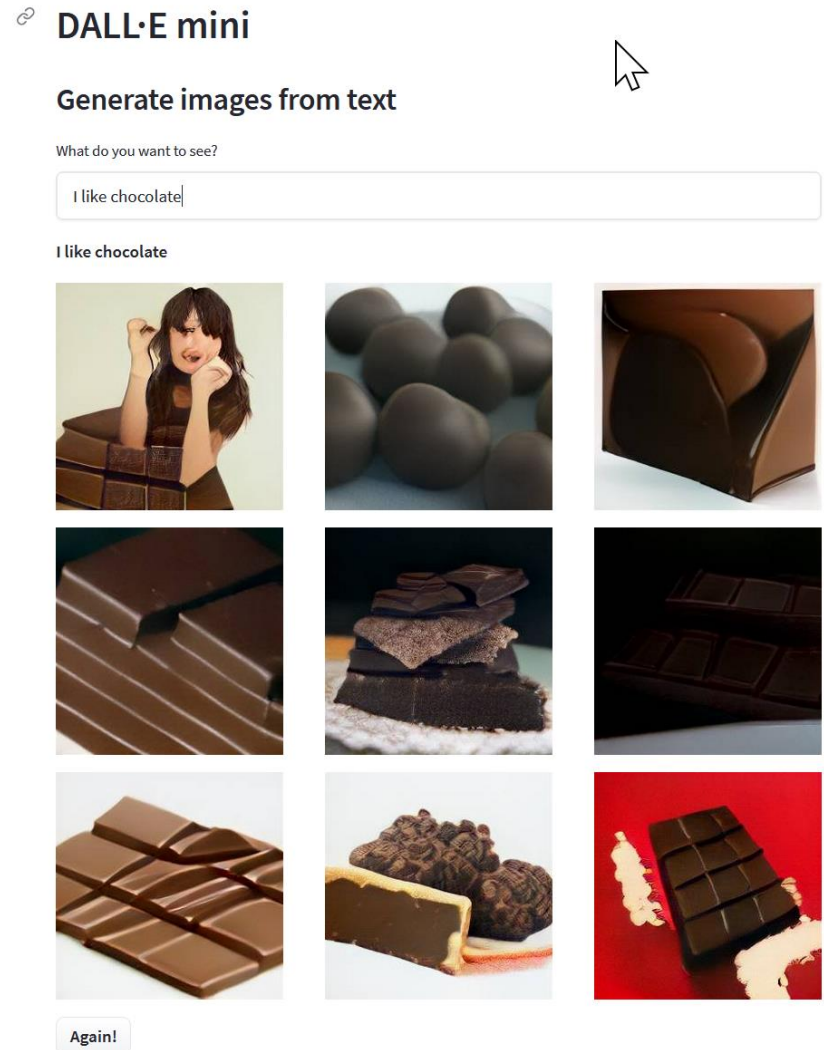
## AIアシスタントの主な用途

- 音声による家庭内装置の操作
- スマートフォンの音声コマンド
- スマートスピーカーの操作
- 日常的なタスク支援

# マルチモーダルAI（テキスト，画像の連携）

## テキストと画像の連携例：

- 入力：  
テキスト "I like chocolate"
- 出力：  
AIが生成した関連画像



AI が生成した画像



## 演習 2. 検索型 AI の体験, チャットボットの体験

# ① Perplexity AI (検索型AI)

- 自然な会話で人間の質問を理解
- 最新の情報を文章で回答.
- 情報源を明示し, 画像を含む質問にも対応可能である.
- 対話を通じて必要な情報を段階的に絞り込むことができる.

## 1. まず Perplexity AI のページにアクセス

<https://www.perplexity.ai/>

## 2. ログインの必要はない

私は原神の初心者です。敵が強くて次に進めません

❖ ソース

原神：敵が硬すぎる時、どうする？ (レベル編) - まどのむこう adymj	原神でまじで敵が強くて困ります。強いというか倒せはする... detail.chiebukuro.ya...	原神で敵が強くて勝てないと感じた時に見て欲しい動画 - YouTube youtube	📱 📺 📺 Show all
---	---	--	-------------------

🌿 回答

原神の初心者で敵が強いと感じる場合、以下の方法で対処できます：

1. キャラクターのレベルを上げる  
キャラクターのレベルを現在の上限まで上げましょう ① ③④。特にメインアタッカーのレベルを優先的に上げることが重要です ①。
2. 武器を強化する  
メインアタッカーの武器を現在の上限レベルまで上げることで、攻撃力が大幅に上昇します ① ③④。
3. 天賦 (スキル) をレベルアップする  
キャラクターの通常攻撃、元素スキル、元素爆発の天賦レベルを上げることで、ダメージが増加します



## ② ChatGPT (チャットボット)

### 1. まず ChatGPT のページにアクセス

ChatGPT

<https://chatgpt.com/>

### 2. ログインの必要はない

※ログインは必須ではありませんが、**Googleアカウントによるログイン**を行うことで、履歴の保存や個人設定などが可能になります。

# (1) ChatGPTデータ分析演習

目的：分析スキルの向上，チャットボット活用スキルの深化

プロンプト例：

**次の売り上げデータを分析してください**

**四半期売上 利益 新規顧客数 製品Aの売上 製品Bの売上**

**Q1 150 30 200 1000 1500**

**Q2 200 50 250 1200 1700**

**Q3 180 40 220 1150 1600**

**Q4 220 60 300 1300 2000**

この売り上げデータを分析すると、いくつかのポイントが浮かび上がります。データは、四半期ごとの売上、利益、新規顧客数、製品Aおよび製品Bの売上に関する情報を提供しています。以下に、各指標の動向を整理し、洞察を示します。

## 1. 総売上の推移

- Q1: 150
- Q2: 200
- Q3: 180
- Q4: 220

## (2) ChatGPTテキストト要約演習



目的：要約スキルの向上，文章理解力の強化

プロンプトの工夫：明確，字数制限の設定，要点の明確化要求

プロンプト例：

**次を100文字以内に要約し要点を明確化してください。**

日本国民は、正当に選挙された国会における代表者を通じて行動し、われらとわれらの子孫のために、諸国民との協和による成果と、わが国全土にわたつて自由のもたらす恵沢を確保し、政府の行為によつて再び戦争の惨禍が起ることのないやうにすることを決意し、ここに主権が国民に存することを宣言し、この憲法を確定する。そもそも国政は、国民の厳粛な信託によるものであつて、その権威は国民に由来し、その権力は国民の代表者がこれを行使し、その福利は国民がこれを享受する。これは人類普遍の原理であり、この憲法は、かかる原理に基くものである。われらは、これに反する一切の憲法、法令及び詔勅を排除する。

日本国民は、恒久の平和を念願し、人間相互の関係を支配する崇高な理想を深く自覚するのであつて、平和を愛する諸国民の公正と信義に信頼して、われらの安全と生存を保持しようと決意した。われらは、平和を維持し、専制と隷従、圧迫と偏狭を地上から永遠に除去しようと努めてゐる国際社会において、名誉ある地位を占めたいと思ふ。われらは、全世界の国民が、ひとしく恐怖と欠乏から免かれ、平和のうちに生存する権利を有することを確認する。

われらは、いつれの国家も、自国のことのみを専念して他国を無視してはならないのであつて、政治道徳の法則は、普遍的なものであり、この法則に従ふことは、自国の主権を維持し、他国と対等関係に立たうとする各国の責務であると信ずる。日本国民は、国家の名誉にかけ、全力をあげてこの崇高な理想と目的を達成することを誓ふ。

日本国民は、選挙で選ばれた代表を通じて、平和と自由を守り、戦争の惨禍を避け、国際協調と平和維持を目指すことを誓い、主権が国民にあることを宣言する。

# (3) ChatGPT プログラム作成支援

目的：プログラミングスキルの向上

プロンプトの工夫：明確，言語の指定

プロンプト例：

## Python で西暦からうるう年を求めるプログラム。分かりやすい解説付き

うるう年 (Leap Year) は、通常の年より1日多い366日がある年です。うるう年を判定するためのルールは次の通りです：

1. 西暦が4で割り切れる年はうるう年。
2. ただし、西暦が100で割り切れる年はうるう年ではない。
3. ただし、100で割り切れても、400で割り切れる年はうるう年。

このルールに基づいて、うるう年を判定するPythonのプログラムを作成します。

### Python プログラム (うるう年判定)

```
python 📄 コードをコピーする

def is_leap_year(year):
    # 4で割り切れる年はうるう年
    if year % 4 == 0:
        # 100で割り切れる年はうるう年ではない
        if year % 100 == 0:
            # 400で割り切れる年はうるう年
            if year % 400 == 0:
                return True
            else:
                return False
        else:
            return True
    else:
        return False

# ユーザーから入力を受け取る
year = int(input("西暦を入力してください: "))

# 判定結果を表示
```

### 解説

1. `is_leap_year` 関数を定義して、うるう年を判定します。
  - まず、入力された年が4で割り切れるかを確認します。
  - 次に、100で割り切れるかを確認します。もし100で割り切れるなら、その年はうるう年ではありません。ただし、400で割り切れる年はうるう年です。
2. `year % 4 == 0` は、年が4で割り切れることを意味します。これが最初のうるう年の判定基準です。
3. 次に、`year % 100 == 0` で、100で割り切れる年かどうかを確認し、もし割り切れる場合、その年はうるう年ではないと判断します。
4. しかし、`year % 400 == 0` の場合は、100で割り切れてもうるう年となります。

### 実行例

```
yaml 📄 コードをコピーする

西暦を入力してください: 2000
2000年はうるう年です。

西暦を入力してください: 1900
1900年はうるう年ではありません。

西暦を入力してください: 2024
2024年はうるう年です。
```

このプログラムでは、ユーザーが入力した西暦がうるう年かどうかを判定して結果を出力します。

## ここまでのまとめ

**自然言語処理は、人間が普段使う言語（日本語、英語など）をコンピュータが理解したり、生成する技術**

### 応用例

- **情報検索**：インターネットから情報を検索
- **AIとの対話、AIへの指示**：AIが対話や指示に応じる
- **プログラミング支援**
- **人間の指示による文書の作成や推敲**
- **翻訳**：異なる言語間で翻訳
- **要約**：文章から要点を抜き出す

## 11-3. 単語ベクトル

# 単語ベクトルの概要



**単語ベクトル**は、自然言語処理の基礎技術

- 単語の数値化
- 意味的類似性の計算を可能にする

## 語の類似性を扱う際の課題

私の母は**親切だ**

私の母は**優しい**

私の母は**思いやりがある**

これらの単語の類似性を，データに基づいて裏付けることが重要（AI の有効分野の1つ）





# 数値と距離



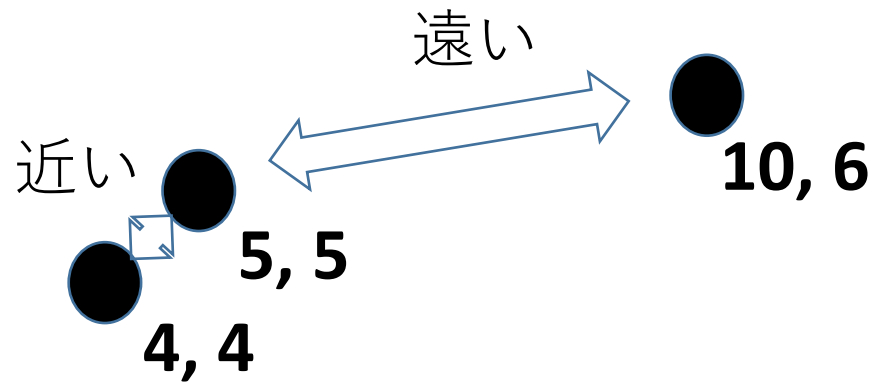
近い  
(距離: 1)

遠い  
(距離: 90)

この考え方を多次元（例：300次元）に  
拡張して単語の類似度を表現する

# ベクトルの距離

2次元（2個の数値）での例



- 近い点 :  $(4, 4), (5, 5)$
- 遠い点 :  $(5, 5), (10, 6)$

**点同士の距離で類似度を表現することができる。**

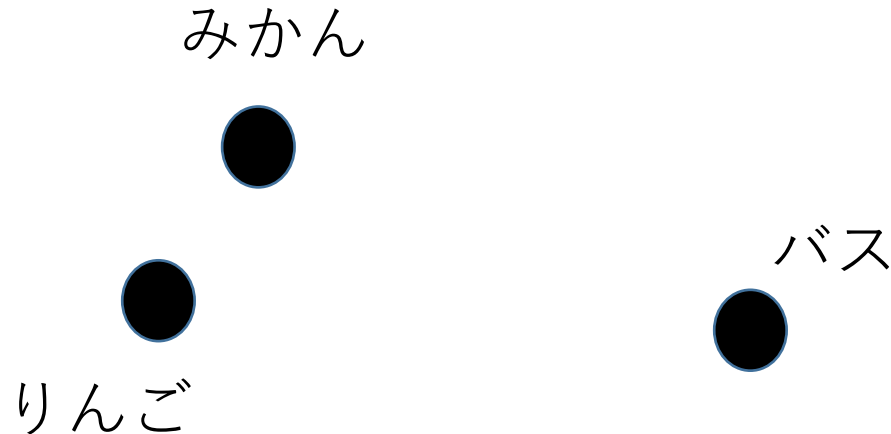
# 単語ベクトルの性質

主な特徴：

- **単語を特徴ベクトル（複数の数値の組）で表現**
- **意味の近い単語は空間的に近く配置**

例：「みかん」と「りんご」は近い

例：「バス」はこれらから遠い位置に配置



# 分布仮説の概念

分布仮説の基本原理：

- **同じ文脈で出現する単語は、類似する意味を持つと仮定**

みかんは甘い

みかんを食べた

りんごは甘い

りんごを食べた

} 互いに  
同じ文脈

バスは速い

バスを探した

} 違う文脈

ヒエログリフの ??? の単語は、何と似ているか？  
 (??? の単語の意味は、実は dog である)

(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

cat, pig  
 と傾向が似ている

値は、2つの単語の共起頻度（共起する回数）

# Word2vec の概要

## Word2vecの特徴

- 2013年に発表された手法
- 分布仮説に基づく学習
- 2つの主要モデル：
  - Skip-Gram：中心語から周辺語を予測
  - CBoW：周辺語から中心語を予測
- 多数の文書データを用いて学習を行う

# Word2vec

- 意味の近い単語は空間的に近くに配置
  - 対立する意味の単語の間に中間的な意味の単語が存在
- 例：「angry」と「pleased」の中間地点に「unhappy」が存在

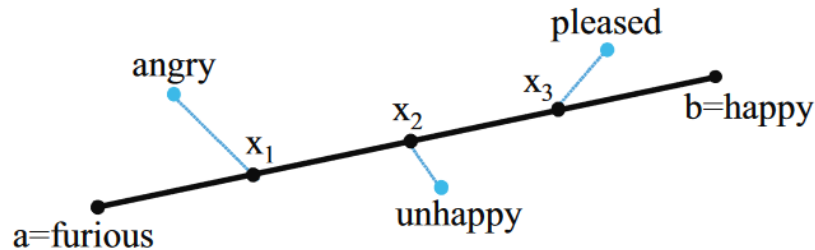


Figure 2: An example of vectors with the highest cosine similarity to intermediate points on the line between *furious* and *happy*.

Joo-Kyung Kim and Marie-Catherine de Marneffe. 2013.

Deriving Adjectival Scales from Continuous Space Word Representations.

In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1625–1630, Seattle, Washington, USA. Association for Computational Linguistics.

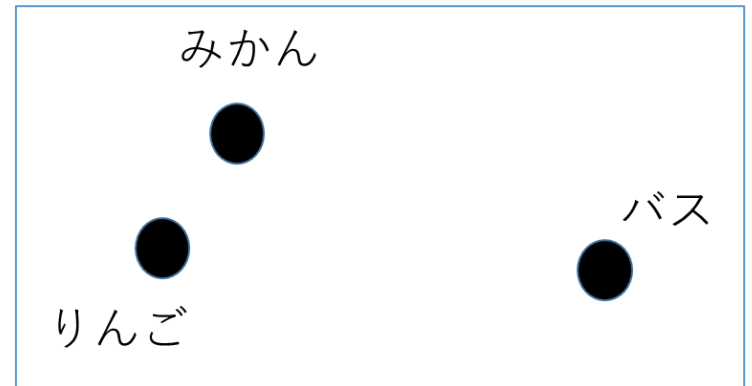


# 単語ベクトルのまとめ

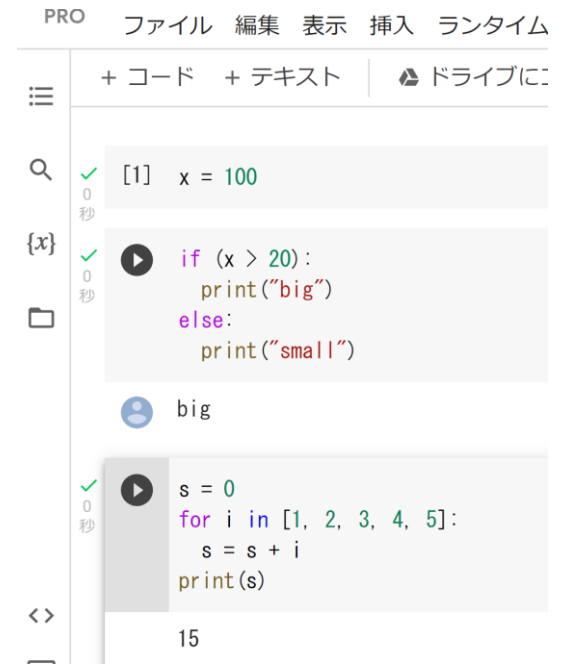
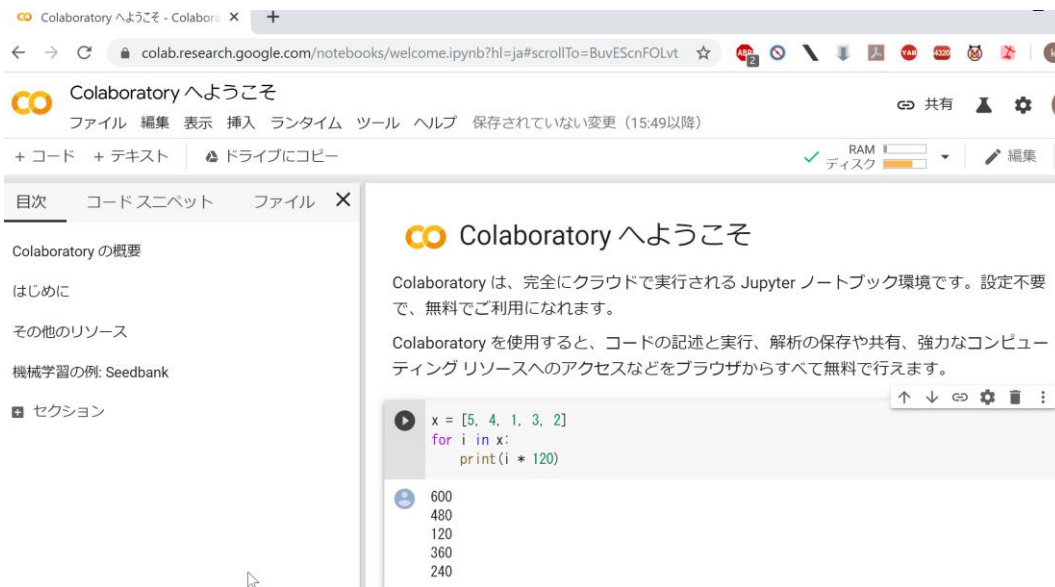
- 語の類似性を考慮
- 単語は特徴ベクトル（複数の数値の組）で表現
- Word2vecは分布仮説と学習に基づく手法。意味の近い単語は空間的に近くに配置



```
[0.4781562 -0.5226188 -0.4748554 0.0077550 -0.2282512 0.1199922  
0.1625752 -0.4287652 0.1621616 0.1165514 -0.2028512 0.1666462  
0.3272642 0.4323962 0.1576176 0.2267312 0.2371740 0.1644210  
0.2816510 0.2282230 0.0899422 0.2382212 0.2295142 0.2624684  
0.1444612 0.2666410 0.2812112 0.4880480 0.0535640 0.2812112  
0.2328114 0.2666750 0.1574112 0.2466170 0.0294200 0.0444114  
0.4900476 0.1827170 0.1742400 0.0466410 0.1112750 0.0466410  
0.0664176 0.1388178 0.0747530 0.1197550 0.2011010 0.1387912  
0.2341112 0.0466410 0.1623400 0.0224200 0.2012112 0.1281960  
0.2722612 0.4664170 0.1762400 0.0535640 0.2666410 0.1811910  
0.3292462 0.2812112 0.1742400 0.1574112 0.1491182 0.1254750  
0.3224170 0.0466410 0.1623400 0.1197550 0.1027544 0.1664180  
0.1024642 0.0224210 0.2282512 0.2266410 0.4884460 0.1109190  
0.0466410 0.2282512 0.1444612 0.2121120 0.1444612 0.0466410  
0.0466410 0.1197550 0.1444612 0.2121120 0.1444612 0.0466410  
0.2666410 0.2282512 0.2812112 0.2417010 0.2722612 0.0466410  
0.1574112 0.0747530 0.4664170 0.1444612 0.1444612 0.2282512
```



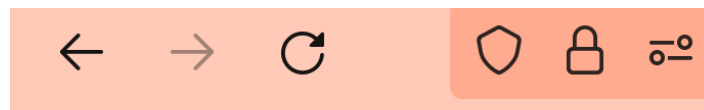
# Google Colaboratory



URL: <https://colab.research.google.com/>

- オンラインで動く
- Python のノートブックの機能を持つ
- Python や種々の機能がインストール済み
- 本格的な利用には, Google アカウントが必要

# Google Colaboratory の全体画面



Colab の定期購入を最大限に活用する  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

メニュー

+ コード + テキスト

コードセル, テキストセル  
の追加



メニュー

(目次, 検索と置換,  
変数, ファイル)

1. 変数

```
[2] x = 100  
    y = 200
```

2. 式

```
▶ print(x + y)  
   print(3 * x + y)
```

300  
500

3. 条件分岐

```
[4] if (x > 50):  
    print('big')  
    else:  
        print('small')
```

big

コードセル,  
テキストセルの  
並び

Web ブラウザの画面

# Google Colaboratory のノートブック

## コードセル, テキストセルの2種類

- **コードセル** : Python プログラム, コマンド, 実行結果
- **テキストセル** : 説明文, 図

2. 式

← テキストセル

```
[5] print(x + y)
     print(3 * x + y)
```

← コードセル

```
300
500
```

3. 条件分岐

← テキストセル

```
▶ if (x > 50):
    print('big')
else:
    print('small')
```

← コードセル

```
big
```



## 演習 3. 単語の特徴ベクトルと単語の類似度

# ① 単語の特徴ベクトル, 単語の類似度

## 1. Google Colabのページにアクセス

[https://colab.research.google.com/drive/1E86OSq90Ayl1z-ULE6hNoVmpCJ\\_7fCR8?usp=sharing](https://colab.research.google.com/drive/1E86OSq90Ayl1z-ULE6hNoVmpCJ_7fCR8?usp=sharing)

## 2. 必要な事前知識

- 単語の特徴ベクトルは, 単語を数値化したもの
- **Word2vec** は, **分布仮説に基づく AI 手法**

## 3. 各自で行うこと

説明の確認

ソースコードの確認

結果の確認

# 11-4. チャットボットの概要

# チャットボットの主な用途



- **翻訳**：言語間の変換
- **校正**：文書の品質向上
- **リサーチ支援**：情報源の探索
- **要約**：テキストの要点抽出
- **プログラミング支援**：バグ発見，コード改善
- **自学発展演習**：質問や相談による学習
- **顧客サービス**：問い合わせ対応
- **エンターテインメント**：コンテンツ作成



# チャットボットの効果



## 期待される効果

- **サービス品質の向上**：24時間即時対応
- **AIと人間の協働**：過去の対応記録の活用
- **エラー検出**：人間が見落とししやすい誤りの発見
- **多言語対応**：言語の壁を超えた対応

# チャットボットの利用上の注意点



## 重要な注意事項

- **不正確な情報**を提供する可能性がある
- **著作権に違反するコンテンツ**を生成するリスクがある
- **プライバシー**や**機密情報**の取り扱いに注意
- 大学の課題などでの**丸写し**は不適切

# ChatGPT の実行例①



- 文章の間違い探し, 文法チェック, 表現の改善, 文章構造の最適化

次の文章から誤字, 不正確な部分を探してください

日本国民は、国家の名誉にかけ、全力をあげてこの崇高な理想と目的を達成することを誓ふ。

この文章の誤字、不正確な部分は以下の通りです。

**誤字：**

- 「かかけ」→「かけ」

「かかけ」は誤字で、「かけ」が正しい表記です。

**修正後：**

- 日本国民は、国家の名誉にかけ、全力をあげてこの崇高な理想と目的を達成することを誓ふ。

また、「誓ふ」という表現は古語的な表現ですが、文脈によっては現代日本語で「誓う」に直す方が一般的です。

# ChatGPT の実行例②



## プログラム作成補助, バグ検出, 改良提案, ドキュメント作成

次はソートを行うPythonプログラムです。改善できますか？ 但し難しい機能は使わないでください。

```
fruits = ['banana', 'cherry', 'date', 'apple']
sorted_fruits = sorted(fruits)
for f in sorted_fruits:
    print(f)
```

あなたのコードはシンプルで十分に動作していますが、いくつかの小さな改善が可能です。コード自体は非常に良く書かれており、難しい機能を使わずに改善する方法として、以下のポイントを考慮できます：

### 1. 変数名の改善：

- `sorted_fruits` という変数名は良いですが、`sorted` という名前は Python の組み込み関数と同じ名前なので、他の名前に変えることでより明確になります。

### 2. 一行でソートして表示：

- `sorted` でソートした結果を直接 `for` ループに渡すことができるので、別の変数に格納せずに一行でソートして表示できます。

これらを反映した改善後のコードを以下に示します。

## 改善案

python

📄 コードをコピーする

```
fruits = ['banana', 'cherry', 'date', 'apple']
for fruit in sorted(fruits):
    print(fruit)
```

# ChatGPT の実行例③



## 自己学習, 自己研鑽

Python で 3次元のゲームプログラミングを開始したい。Godot (Python対応プラグイン)を用いたPythonプログラム, その解説, 実行法, 遊び方, 実行結果を教えてください。単純なものを希望します。

Pythonを使ってGodotゲームエンジンで3Dゲームを開発するためには、まずGodotエンジンにPythonプラグインを導入する必要があります。Godotは、デフォルトでGDScript (Godot独自のスクリプト言語)を使用しますが、Python for Godotというプラグインを使うことでPythonでスクリプトを書くことができます。

以下は、Python対応のGodotを使って、非常にシンプルな3Dゲーム (回転する立方体)を作成する手順です。

### 1. Godotのインストール

まず、Godotエンジンを公式サイトからダウンロードし、インストールします。

- 公式サイト: [Godot Engine](#)

### 2. Pythonプラグインのインストール

Godotエンジンには、デフォルトでPythonがサポートされていません。Pythonスクリプトを使用するために、以下の手順でGodot Pythonプラグインをインストールします。

Pythonプラグインのインストール手順:

1. Godot PythonプラグインをGitHubからダウンロードします。
  - GitHubリポジトリ: [godot-python](#)
2. プラグインをダウンロードして解凍し、Godotプロジェクトの `addons` フォルダにコピーします。
3. Godotエディタでプロジェクトを開き、`Project` -> `Project Settings` -> `Plugins` に進み、Godot Pythonプラグインを有効にします。
4. `python` という新しいスクリプトファイルを作成して、Pythonコードを記述します。

### 3. 新しいプロジェクトの作成

1. Godotエディタを開き、\*\*[New Project]\*\*を選択します。
2. プロジェクトの名前を設定し、保存先を選択します。

### 4. 3Dシーンの作成

1. 新しいシーンを作成します。最初に、`Spatial`ノードをルートノードとして選びます (`Spatial`ノードは3D空間での操作を行う基本ノードです)。
2. 次に、`MeshInstance`ノードを追加し、メッシュのタイプを「Cube」に設定します。これで立方体を作成できます。
3. カメラノードを追加し、シーンを視覚的に確認できるようにします。
4. `Light`ノードを追加して、シーンにライティングを加えます。

### 5. Pythonスクリプトの追加

1. `MeshInstance`ノードを選択し、「Node」タブから「Attach Script」を選択します。
2. 新しいスクリプトとして、`CubeRotation.py` というファイル名をつけ、以下のようにPythonコードを書きます。

### 6. Pythonコード (立方体を回転させる)

```
python 📄 コードをコピーする

from godot import exposed, export
from godot import Node

@exposed
class CubeRotation(Node):
    def __ready(self):
        # 初期化時に呼ばれる
        pass
```

# チャットボットのベストプラクティス



## 実践ポイント

### **プロンプト：**

- 明確で具体的な質問・要求
- 追加要求による回答の改善

### **データや例の活用：**

- 関連データの提供
- 回答として欲しい内容の例示

### **回答の確認：**

- 正確性の検証
- 根拠の確認



## 演習4. チャットボットの活用実践

# 1. まず ChatGPT のページにアクセス

ChatGPT

<https://chatgpt.com/>

# 2. ログインの必要はない

※ログインは必須ではありませんが、**Googleアカウントによるログイン**を行うことで、履歴の保存や個人設定などが可能になります。



# 次の3つのプロンプトを試す

## ①文章の間違い探し

次の文章から誤字、不正確な部分を探してください

日本国民は、国家の名誉にかかけ、全力をあげてこの崇高な理想と目的を達成することを誓ふ。

## ②プログラミング支援

次はソートを行うPythonプログラムです。改善できますか？ 但し難しい機能は使わないでください。

```
fruits = ['banana', 'cherry', 'date', 'apple']
```

```
sorted_fruits = sorted(fruits)
```

```
for f in sorted_fruits:
```

```
    print(f)
```

## ③プログラミング支援

Python で 3次元のゲームプログラミングを開始したい。Godot (Python対応プラグイン)を用いたPythonプログラム, その解説, 実行法, 遊び方, 実行結果を教えてください。単純なものを希望します。

# チャットボットの総括

## 用途

- 翻訳, 校正, リサーチ支援
- 要約, プログラミング支援
- 自学発展演習支援
- 顧客サービス, エンターテインメント など

## 期待される効果

- サービス品質の向上
- AIと人間の協働
- エラー検出の向上
- 多言語対応の実現

## 利用上の注意点：

- 情報の正確性確認
- 著作権への配慮, プライバシー保護への配慮

# LLaVA



- LLaVA はマルチモーダル（言葉、画像という複数を受け付ける）なチャットボットの技術
- LLAVA の公式デモ(llava-v1.5-13b):  
<https://llava.hliu.cc/>

# LLaVA との対話 ①画像の説明



この画像を日本語で説明してください

この画像は、5人のビジネスマンが集まっているオフィスの会議室です。彼らは、それぞれのラップトップコンピュータを使用しています。彼らは、彼らのコンピュータを使用して、ビジネスの議論やプレゼンテーションを行っている可能性があります。

会議室には、植物があり、彼らの周りには椅子が配置されています。また、彼らの周りには、カップやボトルがあり、彼らが飲み物を楽しんでいることが示唆されています。

# LLaVA との対話 ②詳細な説明



この画像で、人間の年齢と性別は推定できますか？

はい、この画像では、人間の年齢と性別を推定することができます。彼らは全員成人であり、男性と女性の比率は50:50です。ただし、具体的な年齢は推定できません。

# LLaVA との対話 ③根拠な説明



この画像が会議室であるという根拠を説明してください

この画像が会議室であるという根拠は、以下の点から推測されます。

1. 人々が集まっている：5人のビジネスマンが一緒にいることから、これは会議室である可能性があります。
2. ラップトップコンピュータの使用：ビジネスマンがラップトップコンピュータを使用していることから、これは会議室である可能性があります。会議室では、プレゼンテーションや議論を行うために、コンピュータを使用することがよくあります。
3. 植物の存在：会議室には植物があり、これは会議室である可能性があります。植物は、会議室の空間を明るく、落ち着かせる効果があるため、よく使われます。

これらの点から、この画像は会議室であるという根拠があります。



## 演習 5 . LLaVA の体験

# ① LLAVAの公式デモにアクセス

<https://lava.hliu.cc/>

- ② 画像を選択（画像を各自で準備することも可能）
- ③ 画像のアップロード
- ④ 質問の入力，対話
- ⑤ 応答の分析



この画像で変なことはありますか？

はい、この画像には変なことがあります。車の荷台には、男性が洗濯機を抱えています。これは、通常の車の荷台に洗濯機を載せることはありません。また、車の荷台には、男性が洗濯機を抱えているという珍しいシーンが描かれています。

チャットボットが画像理解力を示す



- **自然言語処理**：AIによる言語理解，対話，翻訳，文書分析などを可能にする。（活用事例：検索エンジン，チャットボット，機械翻訳など）
- **プロンプトエンジニアリング**：チャットボットに対する効果的な指示・質問の方法のための技術スキル
- **Word2vec**：単語を数値ベクトルに変換する技術であり，分布仮説に基づく手法で，意味の近い単語を空間的に近く配置する。
- **チャットボット活用**：AI技術を用いた対話システムの実践的な利用であり，文書作成，プログラミング支援，学習支援などの多様な用途に活用可能。（注意点：情報の正確性確認，著作権への配慮，プライバシー保護）