3章
自然言語処理入門

01

自然言語処理で何ができるのか？

前章では、AIや機械学習の仕組みについて理解を深めました。本章は生成AIへの序章として、自然言語処理について紹介します。自然言語処理はNLP（Natural Language Processing）とも呼ばれ、我々が使う自然言語（日本語や英語といった話し言葉）で書かれた文章をAIが読み取り、実行する処理です。自然言語処理の技術は、テキストの解析から意味の理解、さらには生成まで、幅広い領域にわたっています。この分野の発展には、統計学的手法や機械学習、最近では深層学習が大きく貢献しています。自然言語を処理することにより、AIは質問に答えたり、文章を要約したり、さらには人間の言葉で新しい文章を生成することが可能になります。 自然言語処理(NLP)は大きく自然言語理解(NLU)と自然言語生成(NLG)に大別されます。

- 自然言語理解（NLU：Natural language understanding）
  - 自然言語理解は、AIが人間の言葉を理解し、その意味や文脈を把握するプロセスです。形態素解析、構文解析、文脈解析などを行って文章の内容を理解する処理であって、文章中の意図や情報を正確に解釈することを目的とします。NLUのアプリケーション例としては、文章分類、文脈認識、感情分析、情報抽出などがあります。ユーザーが入力した自然言語のプロンプト（クエリを含む）に基づいて正確な情報を提供したり、テキストから特定の情報を抽出したりするのに役立ちます。
  - Google検索エンジンのアルゴリズムに使用されるようになったBERTが有名です
- 自然言語生成（NLG：Natural Language Generation）
  - 自然言語生成は、入力された文章の続きを生成したり文章の変換を行う処理であって、AIが自然言語で新しいテキストを作り出すプロセスです。NLGの用途として、機械翻訳、要約文生成、AIによる創作活動（例えば、物語や詩の作成）などがあります。
  - OpenAIのGPTなどが有名です

簡単にまとめると、文章から文脈や意味合いを抽出する処理がNLUであり、文脈や意味合いから文章を生成する処理がNLGとなります（図X-X）。

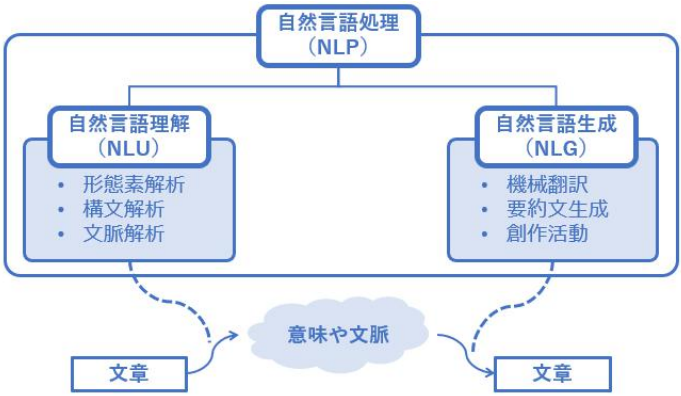


図 X-X 自然言語処理

自然言語理解（NLU）の使いどころ

音声認識

スマートフォンやスマートスピーカーなどのデバイスでは、NLU 技術を用いてユーザーの発話からコマンドを認識し、音楽の再生、ニュースの読み上げ、天気予報の提供、スマートホームデバイスの制御などを行います。多少の表現の揺らぎがあっても、ユーザーが実行したい処理を意図通りに実行したり、知りたい内容に対して回答する事が可能になります。

情報抽出と知識管理

NLU 技術は、大量の文書やテキストデータから重要な情報を抽出し、構造化するのにも用いられます。インターネット検索はもちろん、企業において契約書、報告書、学術論文などから必要な情報を迅速に見つけ出し、知識の管理と活用を効率化が可能となります。

感情分析

NLU の応用例として、感情分析があります。これは、文章に含まれる感情や意見を識別する技術です。ソーシャルメディアの投稿、製品レビュー、顧客フィードバックなどから、ポジティブ、ネガティブ、ニュートラルなどの感情を自動で判定します。企業は感情分析を活用して市場のトレンドを把握したり、顧客満足度を評価することが可能となります。

自然言語生成（NLG）の使いどころ

機械翻訳（MT: Machine Translation）

ある文章に対して自然言語処理を行い、別の言語に変換するものです。生成 AI 技術は翻訳ツールとして開発されたものではありませんが、ある文章を特徴量空間に埋め込み、別の言語として取り出す[1]という所作を行うことによって、翻訳でも高い能力を発揮します[2]。

コンテンツ生成

特定のデータに基づき、人間にとって読みやすい概要を自動で作成する応用例です。近年では Google 検索結果に生成 AI による検索体験 (SGE: Search Generative Experience) が追加されています。同社は、目的の情報をすばやく簡単に見つける新たな方法と位置づけています[3]。

図 X-X SGE の例

---

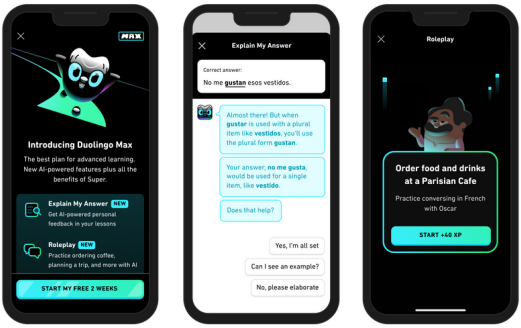
[1] 特徴量空間に埋め込む話は第一章でも紹介しました。

[2] 例えば 2023 年に公開された論文（https://doi.org/10.48550/arXiv.2301.08745）を紹介します。この論文は 2023 年 1 月に第 1 版が公開され、その際には「データが豊富なヨーロッパ言語等においては商業翻訳製品（Google 翻訳など）と競合する性能を発揮する一方で、データが少ない言語や遠い言語では大幅に遅れをとっている」旨の結論でした。その後に ChatGPT-4 が公開されたことによって論文も更改され、2023 年 11 月に公開された第 4 版では「GPT-4 では翻訳性能は大幅に向上し、遠い言語であっても商業翻訳製品と遜色ない性能である」旨の結論が述べられています。なお、当該論文における翻訳性能の評価には BLEU スコアが使用されています。BLEU スコアは機械翻訳結果の精度評価指標として広く利用されており、自動翻訳と人間が作成した参考翻訳との違を測定するものです。参考翻訳文と同一の単語列を含む出力文が高いスコアを算出するため、出力文において文法的な誤りが存在しても高いスコアを算出してしまうという欠点もあり、人間が翻訳文を評価した場合には異なるスコアが算出される場合があります。GPT4 が人間の翻訳者を完全に置換でき得るかという点については注意して考える必要があるでしょう。

[3] 同機能は 2023 年 8 月から日本国内でも試験提供が開始されました（https://japan.googleblog.com/2023/08/search-sge.html）。

チャットボットと仮想アシスタント

チャットボットについてはChatGPTを想像するとよいかもしれません。自然言語生成（NLG）技術を利用して、人間との会話をシミュレートする応用例です。 世界で1億人以上のユーザーを有する言語学習サービス「Duolingo」は2023年にOpenAI社のGPT-4を採用し、言語学習者が対話型AIチャットボットを通して会話力を鍛える機能を提供しています。学習者はチャットボットからフィードバックや励ましの言葉を受け取ることもできます[4]。



図X-X 言語学習チャットボットの例

どのように機械学習すればいいの？

自然言語処理は我々の身近なアプリケーションに広く応用されているというイメージを持っていただけたと思います。ここで、第二章で学んだように、あらゆる機械学習モデルは数字で処理を行います。自然言語を何らかのカタチで数字表現に変換する必要があります。次節では、文章を数字表現に変換する技法を紹介します。

[4] 同機能は2023年3月に「Duolingo Max」として公開されました（https://blog.duolingo.com/duolingo-max/）。一方で2023年時点では一部の言語に限り提供されており、米国や英国などの一部地域でiOS版のみの提供となっています。

02 離散化 ～文章を区切る技術～

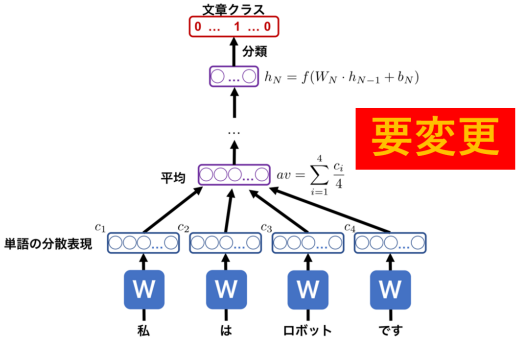
第一章で言及した通り、自然言語から成る文章は本質的に連続的で複雑なデータであり、時に長い文脈を持ち、一部を切り出しただけでは意味をなさないこともあります。一方で、機械学習モデルで自然言語を処理する際には、モデルが受け取れる形式で受け渡さなければなりません。すなわち、固定長の数字データ[5]に変換し、コンピューターが処理しやすい形式に変換（離散的な要素に分割）する必要があります。

文章を区切る

文章を数字に変換するにあたって、まず文章を適当に区切ることが必要です。文字を適当な単位に区切り、この各単位を特徴量とすれば、先に学んだ機械学習で処理ができそうです。このように適切な単位に分割する事をトークナイズ（tokenize）[6]と呼びます。英語の場合は、各単語はスペースで区切られているので機械的に処理することが可能ですが、日本語の場合は明示的な区切り記号がないので工夫が必要です。 文字ごとに区切ってしまう方法（character 分割）もありますが、例えば「生成」という単語はこれ単独で"creation"という意味を持ちますから、「生」（"raw"）と「成」("consist"ないし"compose"）という二つに分割しない方が、文章中の本来の意味を保持する事が出来そうです。

[5] 第二章で学んだ通り、機械学習モデルは幾つかの特徴量を基にして推論を行います。例えば「来場者数」と「気温」という2種類の特徴量から「売上」を予測するように学習したモデルに対して、推論段階で「湿度」という新たな特徴量を追加して推論を実行することはできません。様々な長さの文章がありますが、文章の長さに応じてモデルの中身を変えるわけにはいかないので、任意の文字列を所定の長さの数字（固定長の数字データ）に変換します。

[6] 日本語では「分かち書き」とも。BERTやGPT等の機械学習モデルでは、このような処理をエンコード（encode）と表現されます。



図X-X 文章を区切るイメージ

形態素解析

図X-Xの自然文章（日本語）を考えた時に、要素ごとに空白で区切りを入れます。この各要素のことを形態素（morpheme; 言語で意味を持つ最小単位）と呼びます。端的に言えば、語彙や文法をルール化し、「少女」は名詞といった具合で、ルールに基づいて分割を行います[7]。 形態素を品詞単位で区切ると、例えば名詞や形容詞といった特定の品詞だけ抜き出して分析を行ったり、係り受けにより文章を解析することができそうです。係り受けとは「文中の言葉同士の関係性」の事で、係り受け解析とは文中の各要素（単語ないし文節）がどのように関連し合っているかを解析する技術を指します。これは、文を構成する要素間の「係り受け関係」を明らかにすることで、文の意味構造を理解するための重要な手段です。日本語では動詞や助詞などが、文節間の関係性を示す重要な因子となり、これらを解析することで文の全体的な意味を捉えることができます。 例えば「サングラスをかけた少女が公園を駆け回る」という文章において、「少女が」が「駆け回る」に係る主語であること、「公園を」が「駆け回る」に係る目的語であることなど、各文節がどのように他の文節に関連しているかを解析します。この解析により、文の構造を理解し、より複雑な自然言語処理タスクへの応用が可能となります[8]。

[7] 最もシンプルな手法は辞書のようにルールを定義するやり方なのですが、2000年前後からは日本語文書の単語出現頻度を考慮して機械学習を行う形態素解析手法（Mecab、Kuromoji、JUMAN、Chasen、など）が現れました。昨今は深層学習を利用した手法（JUMAN++、など）も提案されています。

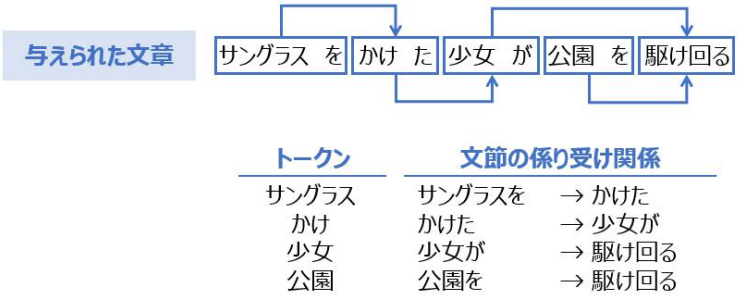


図 X-X 係り受け解析

サブワード分割

先ほどのように形態素や文節で区切ってしまえば離散化自体は可能そうですが、世の中にある全ての形態素を列挙することは非現実的です[9]。更に、学習時に登場しなかった未知語に対しては上手く対処できません[10]。こういった背景から考案されたのが、単語を更に小さな単位（サブワード）に分割しようというサブワード分割です。極端な話ですが、英語文章を全てサブワードに分解すると、アルファベット 26 種になります。ここまで分割すると、全ての英文は 26 種の語から構成されることになりますので、語彙数（Vocaburary）を圧縮できます。論文 1 本でも学習させれば、全アルファベットが使用されているでしょうから、基本的に未知語に遭遇することはなくなるでしょう。 実際には、機械学習によって適切なサブワード単位が決定されます。

形態素解析器により入力文を単語単位に分割し、分割された各単語に対して、サブワード分割手法（BPE[11]、WordPiece[12] など）を用いてサブワード単位に分割します。れらの手法は主に教師なし学習が用いられていて、文章

[8] 図 X-X 中の係り受け解析には自然言語処理ライブラリ「GiNZA」（ギンザ）を用いました。「GiNZA」は株式会社リクルートの AI 研究機関である Megagon Labs と国立国語研究所との共同研究により生まれた日本語自然言語処理オープンソースライブラリです。先進的な自然言語処理ライブラリ「spaCy」をフレームワークとして使用しており、日本語の形態素解析器「SudachiPy」を使用しています。（https://www.recruit.co.jp/newsroom/2019/0402_18331.html）

[9] 仮にこのようなことが出来たとしても、ほぼ無限の語彙数（vocaburary）が必要となり、計算量が増えてしまいます。

[10] 例えば、「マイナポイント」という語句が辞書に登録されていないと「マイナ」「ポイント」（これは 2013 年に公開された"unidic-mecab-2.1.2"という辞書を用いた場合の分割例です。マイナポイント事業が開始されたのは 2019 年でしたので、辞書作成当時は未知語でした）といった具合で、意図せずに分割されてしまうことが考えられます。日本語形態素解析における未知語処理手法として、既知語からの派生ルールを用いる方法などが提案されていたりもします（https://doi.org/10.5715/jnlp.21.1183）。

データさえあればトークナイザを構築できます。サブワード単位に分割された語を言語モデルへの入力に用いることで、低頻度語に対して頑健性を向上させます。

SentencePiece

一方、それなら最初から形態素に分けずに分割してしまおう、という手法がSentencePieceです。この手法は言語に関する事前知識を必要とせず、生のテキストデータから直接サブワード単位を学習します。英語圏言語で用いられている空白スペースも文字として扱うので、日本語や中国語のようにスペース区切りでは無い言語と同様に扱うことが可能です。 Googleが開発し、GitHub上でオープンソースとして公開[13]されているため、研究者や開発者は自由に利用できます。 SentencePieceは、特にトランスフォーマーベースのモデル（例えば、BERTやGPT）の前処理として使用されています。

まとめ

形態素による係り受け解析は、助詞や名詞のデータベースに基づく、人が作成したルールベースの処理によって文章を解析します。自然言語の文法的構造を理解する上で重要であると考えられています。このようなルールベースのアプローチは、特定の言語の文法規則や特性を詳細に反映することができるため、高い精度での解析が可能です。一方でこの手法は、ルールの作成とメンテナンスに多大な労力が必要であり、未知の表現や言語の変化に対応するためには定期的な更新が必要となります。

BERTやGPT等の現代的な機械学習モデルでは、Encode後のトークン間の関連性を学習して、人が作成したルールに依存せずに自ら係り受けの学習（に近いこと）をしていると考えることができます。このように自ら特徴を学習するには、Self-Attentionという要素技術が重要なのですが、こちらについ

[11] BPE（Byte Pair Encoding; バイトペア符号化）とは、文字の繋がりを見て、頻出する文字同士を結合させる手法です。要するに、単語内の繋がりやすいアルファベットの組み合わせをサブワードとするものです。日本語においては元々の語彙数がアルファベットよりも段違いに多いので、そのペアを探して登録する事を繰り返すと、語彙数が逆に大くなってしまうことが知られています。
[12] 考え方としてはBPEに近いです。WordPieceは最も頻度の高いシンボルペアを選択するのではなく、一度語彙に追加された学習データの尤度を最大化するものを選択します。
[13] GitHubはソフトウェア開発のプラットフォームで、個人や企業を問わず幅広いユーザーがコードを共有しています。8000万件以上ものプロジェクトが管理されており、SentencePieceは次のURLで Apache2.0 ライセンス下で公開され、尚も継続的に開発とメンテナンスが行われています（https://github.com/google/sentencepiece）。

ては Transformet に関する説明の章で述べます。 OpenAI の GPT シリーズでは、tiktoken というトークナイザー[14] が使用されています。図 X-X に「こんにちは。今日の天気は？」を分解した様子をしまします。「こんにちは」のように語句は単一のトークンとして処理されていますが、それ以外の文字は単独で区切られています[15]。

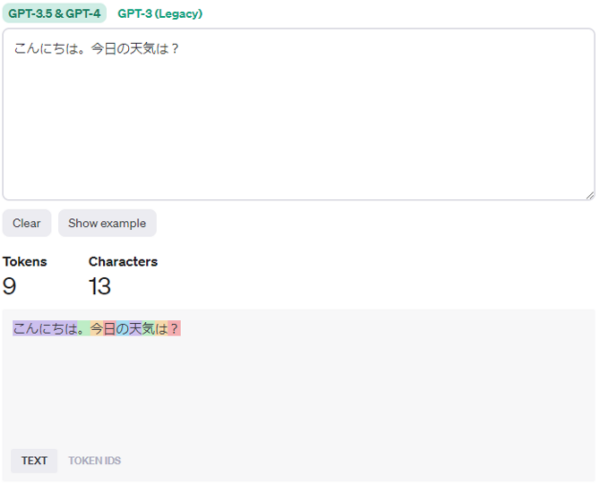


図 X-X GPT における文章の区切り方

人間の感覚からすれば、「天気」は単独で取り出した方が意味を成しそうですが、多言語の大量文章を学習した結果、これらは分けた方が都合が良いという結果になったのでしょう。現代的な機械学習モデルで使用している Self-Attention が強力なのは、人が作成したルールに依存せずに、広域的な単語（Encode した後のトークン単位）間の関連性を学習出来る点です。例えば係り受けを学習するにあたって、人間が作成したルールに基づいて分割された単語間の係り受けを学習する限り人間の性能を超えることは難しいでしょう。

[14] 高速 BPE トークナイザーで、トークナイザーの挙動は OpenAI 社の Web ページ上で確認することが出来ます（https://platform.openai.com/tokenizer）。
[15] GPT-4 や GPT-3.5 では、「cl100k_base」というエンコーディングが使用されています。例えば、"こんにちは。今日の天気は何ですか？"という文章をトーカナイズすると、次の 12 トークンに分割されます。 トークン：['こんにちは', '。', '今', '日', 'の', '天', '気', 'は', '何', 'です', 'か', '？'] これらの各トークンには ID が割り振られており、モデルに入れる際には文字ではなく、これらの ID で識別されることになります。（トークン ID： [90115, 1811, 37271, 9080, 16144, 36827, 95221, 15682, 99849, 38641, 32149, 11571]）

ルールベースの形態素解析ではなく、更に細かいトークン単位での学習を行う大きな理由です。 Self-Attention メカニズムを採用した現代的な大規模言語モデルは、単語やフレーズの分割方法に対する人間の直感的な理解を超える能力を持っています。これらのモデルは、大規模なデータセットから複雑な言語パターンを学習することで、より洗練された言語理解と処理を実現します。この進歩は、人間が作成したルールベースのシステムでは到達できない新たな可能性を開くものであり、自然言語理解の分野における今後の研究や応用に大きな影響を与え続けるでしょう。

参考：前処理手法

ここでは、参考として文章を離散化する際に用いられる前処理手法を紹介します。対象となる文章に合わせて、複数の手法を組み合わせて適切な分析を行います。

正規化（表記揺れを是正）

文字種の統一、つづりや表記揺れの吸収といった単語を置換して統一する処理をします。後続の処理における計算量やメモリ使用量の観点から見ても重要な処理です。

- 文字種の統一
  - 文字種の統一ではアルファベットの大文字を小文字に変換する、半角文字を全角文字に変換するといった処理を行います。たとえば「Cat」の大文字部分を小文字に変換して「cat」にしたり、「ﾈｺ」を全角に変換して「ネコ」にします。このような処理をすることで、単語を文字種の区別なく同一の単語として扱えるようになります。
- 数字の置き換え
  - 数字の置き換えでは文章中に出現する数字を別の記号(たとえば0)に置き換えます。たとえば、ある文章中に「2017年1月1日」のような数字が含まれる文字列が出現したとしましょう。数字の置き換えではこの文字列中の数字を「0年0月0日」のように変換してしまいます。数値表現が多様で出現頻度が高い割には自然言語処理のタスクに役に立たないことが多いからです。
- 辞書を用いた単語の統一

- 辞書[16]がを用いた単語の統一では単語を代表的な表現に置き換えます。たとえば、「オレンジ」と「orange」という表記が入り混じった文章を扱う時に、どちらかの表現に寄せて置き換えてしまいます。これにより、これ以降の処理で2種類の単語を同じ単語として扱えるようになります。置き換える際には文脈を考慮して置き換える必要があることには注意する必要があります。

ストップワード（stop word）

ストップワードとは、自然言語処理を実行する際に一般的で役に立たない等の理由で処理対象外とする単語のことです。たとえば、助詞や助動詞などの機能語(「は」「の」「です」「ます」など)が挙げられます[17]。これらの単語は出現頻度が高い割に役に立たず、計算量や性能に悪影響を及ぼすため除去されます。 ストップワードの除去には様々な方式がありますが、この記事では以下の2つの方式を紹介します。

- 辞書による方式
  - この方式では、あらかじめ定義されたストップワードのリスト（辞書）を使用して文章からストップワードを除去します。この辞書は、その言語で一般的に使用される前置詞、冠詞、接続詞などの単語を含んでおり、自然言語処理においてほとんど意味を持たないと考えられる単語から構成されます。辞書を用いる利点として、シンプルで実装が容易である点が挙げられます。一方で、辞書が固定されているため、特定のドメインやコンテキストに特有のストップワードをカバーしきれない可能性があります。
- 出現頻度による方式
  - この方式では、テキスト全体での単語の出現頻度を分析し、あまりにも頻繁に出現する単語や、逆にほとんど出現しない単語をストップワードとして扱います。頻繁に出現する単語は、一般的に内容の理解にあまり寄与しないと考えられるため、ストップワードとして除外されます（例えば、「の」「あの」「この」など）。一方、非常に稀にしか出現しない単語も重要な意味を持たない場合が多いため、除去の対象となることがあります。与えられたデータに対して柔軟にストップワードを定義できますが、単語の

[16] 有名な辞書としてWordNetが挙げられます。単語間の意味的関係を体系的に整理した大規模な辞書データベースで、単語を「シノニムセット（synsets）」と呼ばれる同義語のグループに分類し、これらのグループ間の様々な意味関係（例えば、上位語/下位語関係、部分/全体関係など）を定義しています。異なる単語が同じ概念を指している場合や、同じ単語が複数の意味を持つ場合、WordNetを使用してこれらの単語や意味の関係を特定し、文脈に応じて表記ゆれを是正することが可能です。
[17] 英語のストップワードとしては、"the"、"a/an"、"is"、"at"、"which"、"on" などの前置詞、冠詞、接続詞が挙げられます

出現頻度だけを基準にして機械的に除去してしまうと、重要な情報を持つ可能性のある単語を除去してしまうリスクも伴います。

まとめ

自然言語で書かれた文章の前処理では、正規化やストップワードの除去といったステップを適切に行うことが、分析精度を大きく左右します。文章データから、分析に不要な表現の揺らぎを取り除き、データを機械学習モデルやその他の分析手法で扱いやすい形に整形するために不可欠です。どのような前処理手法を適用するか、またその程度は、次ような観点に基づいて慎重に選択する必要があります。

- 文章データの性質
  - 文章で扱われている言語や専門分野によって、適切な前処理手法が異なります。例えば、技術文書や医療記録などの特定のドメインでは、稀有な専門用語であっても重要な意味を持つと考えられるため、これらをストップワードとして除外すべきではないでしょう。
  - SNSの投稿、ニュース記事、学術論文など、テキストの形式や出典によって、言語の使用法やスタイルが異なるため、前処理のアプローチも変わってきます。例えば、絵文字や顔文字は、その感情的な表現に意味があったり、何かを代替して表現する場合があるため、どのように処理を行うかは検討に値します。
- 分析の目的
  - タスク: 分析の目的に応じて、特定の単語やフレーズが重要になったり、逆に無視されたりします。例えば、感情分析では否定語が重要な役割を果たすため、これらを除外しないようにする必要があります。
  - モデルの要件: 使用する機械学習モデルやアルゴリズムによって、データの形式やどの情報が重要であるかが異なります。深層学習モデルでは生のテキストデータをそのまま利用できることもありますが、統計的モデルではより細かい前処理が必要になる場合があります。

03 単語文章行列（BoW, TF-IDF）

前節では文章をトークンに分割するイメージについて理解を深めました。一方でこのままでは尚も数字ではないため、数値的に解析可能な形式に変換するため（モデルに入力するため）には何らかの処理が必要そうです。 各トークンに対して、例えば文書に出現する単語の頻度を数値化すれば、各トークンの特徴を表現できそうです。このように作成したデータを単語文書行列と言い、文章を数値化する上で基本的な手法となります。

Bag of Words (BoW)

BoW モデルは、テキスト内の単語の出現回数をカウントし、それを特徴ベクトルとして表現します。このとき、単語の順序や文脈は無視され、単純に単語の「袋」として扱われます。例えば、二つの文書があった場合、BoW モデルは各文書に含まれる全ての単語のリストを作成し、各単語がその文書に何回出現するかをカウントします。この方法により、文書を固定長のベクトルとして表現することができ、テキスト分類や検索、さらには感情分析など、多岐にわたるアプリケーションで利用されます。

TF-IDF

TF-IDF は、BoW モデルをさらに洗練させた手法で、単語の出現頻度だけでなく、その単語がどれだけ「重要」であるかを加味した重み付けを行います。具体的には、TF（term frequency）は単語が文書内にどれだけ頻繁に出現するかを示し、IDF（inverse document frequency）はその単語が文書集合全体の中でどれだけ珍しいかを示します。これにより、一般的な単語（例えば「は」「が」「と」など）が高い頻度で出現しても、その重要度は低く評価され、特定のトピックに特化した単語が強調されます。TF-IDF は、情報検索や文書分類、文章の類似度計算などに広く用いられています。

これらの手法により、テキストデータから有用な特徴を抽出し、機械学習モデルの入力として利用することができます。BoWとTf-idfは、テキストデータを扱う上での基礎であり、これらの概念を理解することは、生成AIを含む多くの自然言語処理アプリケーションの背後にあるメカニズムを理解する上で欠かせません。

04 word2vec (Skip-gram, CBOW)