統計情報で社会・経済を診断する

第10回

～文章から社会・経済を診断する～

今日の内容

- テキストマイニングの方法について学習する。日本語を形態素解析するMecabの使用方法、Zipfの法則、タイプ、トークン比、TF-IDFなど基礎知識の習得を目指す。

テキストマイニング実例

朝日新聞デジタル > 記事

2022参院選 ニュース 開票速報 注目の当確 開票ライブ 企画連載 ボートマッチ 朝日・東大調査 動画

最後に迫力アップの演説　聴衆が去らない演説　変化をデータで分析

有料記事 参院選2022

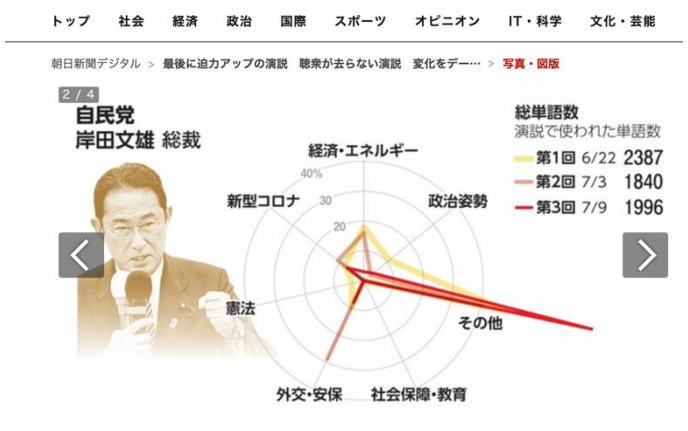
高木智也、松山紫乃、高橋杏璃 鬼原民幸 小野太郎 小手川太朗 横山翼 藤崎麻里 阿部彰芳 森岡航平 里見稔 2022年7月12日 7時00分

list

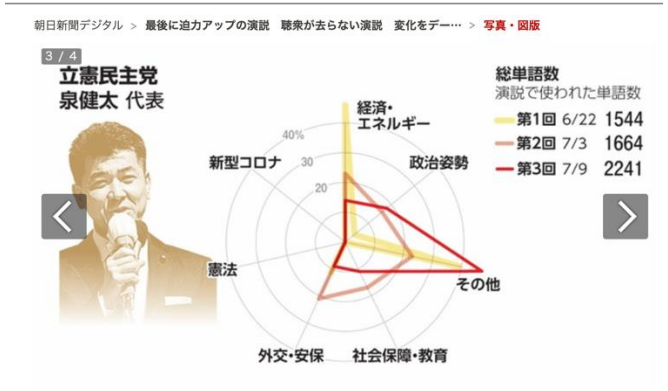
グラフィック・荻野史杜

18日間の 参院選 が終わった。9党首は何を訴え、訴えはどう変化したのか。朝日新聞は各党首の演説を1カ所ずつを取り上げ、自然言語処理技術を使って分析。6月

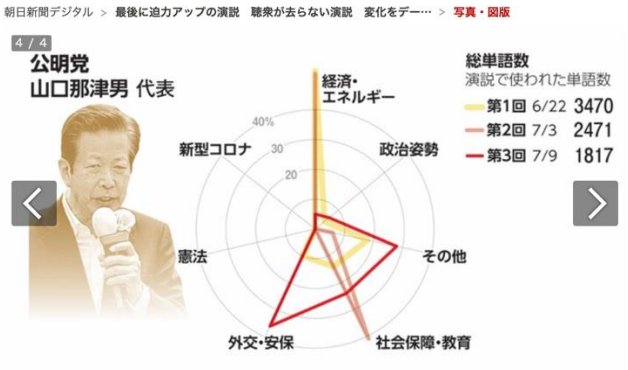
https://www.asahi.com/articles/ASQ7C63YBQ7CUTFK015.html?iref=comtop_7_01



自民党の岸田文雄総裁



立憲民主党の泉健太代表



公明党の山口那津男代表

テキストマイニング実例

寒さに関連した会話量

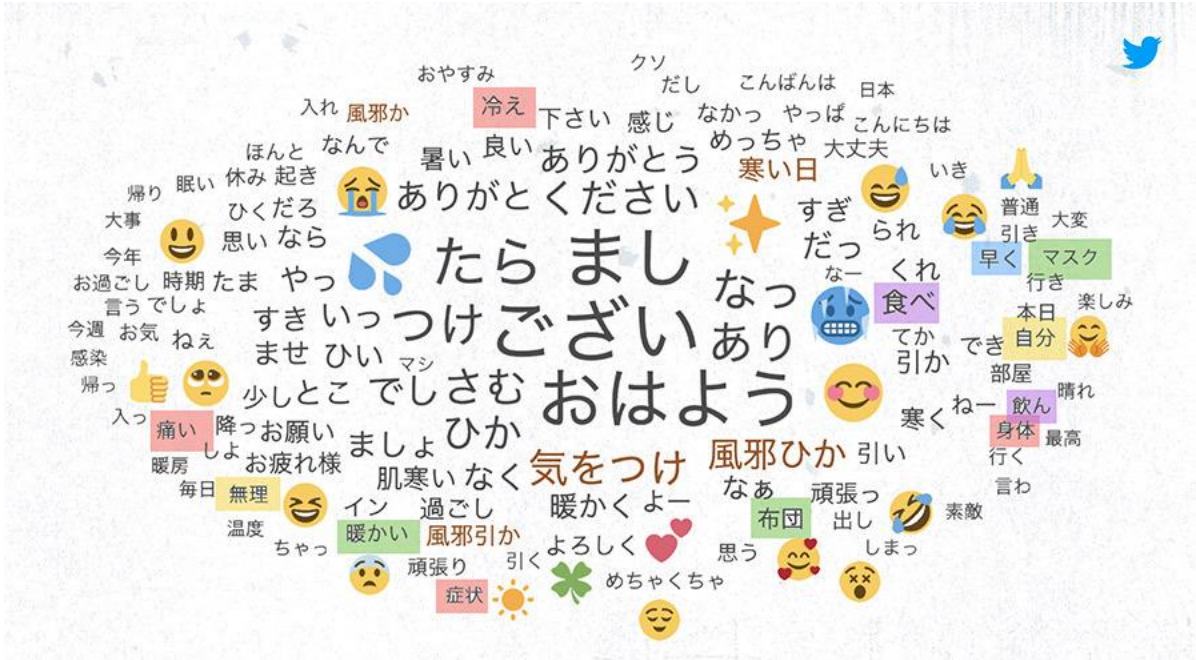
Source | Brandwatch, 2021年1月1日-2021年12月31日までのツイート量より算出。対象キーワード：寒い、さむい、風邪

寒さに関連した会話が増え始めるタイミング

	会話が増え始めるタイミング	その時期の最低気温（東京）
2017年	10月前半	19°C
2018年	11月後半	6°C
2019年	10月前半	16°C
2020年	10月前半	20°C
2021年	10月前半	19°C

Source | Brandwatch, 2021年1月1日-2021年12月31日までのツイート量より算出。対象キーワード：寒い、さむい、風邪

- 冬の寒さと「コネクト」する - 2022年
  - https://marketing.twitter.com/ja/insights/connect-with-whats-happening-cold-winter-2022



形態素解析とは？

- 形態素：意味を持つ最小の単位
  - 例：音、教える（教え＋る）
- 形態素解析：文を形態素ごとに分割し、品詞情報などをつけ加える作業。

主な形態素解析エンジン

ツール	開発年	開発	主な特徴・用途
Juman	1992年	京都大学	高精度な形態素解析。学術研究や構文解析に活用。後継ツールにJuman++がある。
ChaSen	1997年	松本 裕治(NAIST)	Jumanをベースに開発された。MeCabの元となる。茶筅。
MeCab	2002年	工藤 拓	高速・軽量な一般的日本語解析ツール。ChaSenを基に開発。RMeCabでR対応。
CaboCha	2000年	工藤 拓, 松本 裕治	主に係り受け解析（構文解析）に特化。MeCabをベースに開発。
Kuromoji	2010年	Atilika	Java環境で利用可能な解析ツール。LuceneやSolrと統合可能。
KyTea	2009年	京都大学	トークン化、タグ付け、読みの付与に特化。軽量で高速。Kyoto Text Analysis Toolkit。
Janome	2015年	打田智子	Python環境での簡易日本語解析。外部依存が少なく軽量。
Rakuten MA	2014年	楽天	eコマース向け解析。学習ベースの形態素解析ツール。
sudachi	2017年	ワークス徳島人工知能NLP研究所	柔軟性が高く、複合語処理や商用利用にも対応。オープンソース。

MeCab

- 奈良先端科学技術大学院大学出身、現Googleソフトウェアエンジニアで Google 日本語入力開発者の一人である工藤拓によって開発。
- Googleが公開した大規模日本語N-gramデータの作成にも使用された。
- https://github.com/taku910/mecab　（ソースコード）
- 関連：大規模日本語 n-gram データの公開（Posted by 工藤拓・賀沢秀人）

各エンジンの形態素解析の結果

- Juman　「形態素解析について勉強しています。」

形態 けいたい 形態 名詞 6 普通名詞 1 * 0 * 0 "代表表記:形態/けいたい カテゴリ:形・模様"
素 そ 素 名詞 6 普通名詞 1 * 0 * 0 "代表表記:素/そ 漢字読み:音 カテゴリ:抽象物"
解析 かいせき 解析 名詞 6 サ変名詞 2 * 0 * 0 "代表表記:解析/かいせき カテゴリ:抽象物 ドメイン:教育・学習;科学・技術"
に に に 助詞 9 格助詞 1 * 0 * 0 "連語"
ついて ついて つく 動詞 2 * 0 子音動詞カ行 2 タ系連用テ形 14 "連語"
勉強 べんきょう 勉強 名詞 6 サ変名詞 2 * 0 * 0 "代表表記:勉強/べんきょう カテゴリ:抽象物 ドメイン:教育・学習"
して して する 動詞 2 * 0 サ変動詞 16 タ系連用テ形 14 "代表表記:する/する 付属動詞候補（基本） 自他動詞:自:成る/なる"
い い いる 接尾辞 14 動詞性接尾辞 7 母音動詞 1 基本連用形 8 "代表表記:いる/いる"
ます ます ます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます型 31 基本形 2 "代表表記:ます/ます"
。 。 。 特殊 1 句点 1 * 0 * 0 NIL

- MeCab　「形態素解析について勉強しています。」

形態　ケータイ　ケイタイ　形態　名詞-普通名詞-一般　0
素　ソ　ソ　素　接尾辞-名詞的-一般
解析　カイセキ　カイセキ　解析　名詞-普通名詞-サ変可能　0
に　ニ　ニ　に　助詞-格助詞
つい　ツイ　ツク　つく　動詞-一般　五段-カ行　連用形-イ音便　1,2
て　テ　テ　て　助詞-接続助詞
勉強　ベンキョー　ベンキョウ　勉強　名詞-普通名詞-サ変可能　0
し　シ　スル　為る　動詞-非自立可能　サ行変格　連用形-一般　0
て　テ　テ　て　助詞-接続助詞
い　イ　イル　居る　動詞-非自立可能　上一段-ア行　連用形-一般　0
ます　マス　マス　ます　助動詞　助動詞-マス　終止形-一般
。　。　補助記号-句点

形態素解析辞書

- エンジンによって結果が異なるのは、形態素に区切るための情報源となる辞書が違うから。
- ipadic :ChaSen用辞書。
- NAIST-jdic :ChaSen，MeCab用の辞書。
- UniDic :MeCab用の辞書。

集計モデル

- N-gram：n個の隣接した記号の度数を集計する方法
  - unigram: 隣接している記号列の長さが 1
  - bigram:隣接している記号列の長さが2
  - trigram:隣接している記号列の長さが3
- 文字だけでなく、単語、品詞、分節などを単位としても良い。

N-gramの例

- 文字を単位とした場合の例
- きしゃのきしゃがきしゃできしゃする「貴社の記者が汽車で帰社する。」
  - 音声解析や形態素解析の精度を評価するときによく使用される文。

１文字	度数	相対度数
き	4	0.222
し	4	0.222
ゃ	4	0.222
。	1	0.056
が	1	0.056
す	1	0.056
で	1	0.056
の	1	0.056
る	1	0.056
合計	18	1

２文字	度数	相対度数
きし	4	0.235
しゃ	4	0.235
がき	1	0.059
する	1	0.059
でき	1	0.059
のき	1	0.059
ゃが	1	0.059
ゃす	1	0.059
ゃで	1	0.059
ゃの	1	0.059
る。	1	0.059
合計	17	1

３文字	度数	相対度数
きしゃ	4	0.250
がきし	1	0.063
しゃが	1	0.063
しゃす	1	0.063
しゃで	1	0.063
しゃの	1	0.063
する。	1	0.063
できし	1	0.063
のきし	1	0.063
ゃがき	1	0.063
ゃする	1	0.063
ゃでき	1	0.063
ゃのき	1	0.063
合計	16	1

N-gramの例

- 単語を単位とした場合の例
  - 「美味しい寿司を食べる。新鮮な寿司が好きだ。」
  - 助詞は除いて、N-gramを考えると、

unigram	度数	相対度数
美味しい	1	0.17
寿司	2	0.33
食べる	1	0.17
新鮮な	1	0.17
好き	1	0.17
合計	6	1

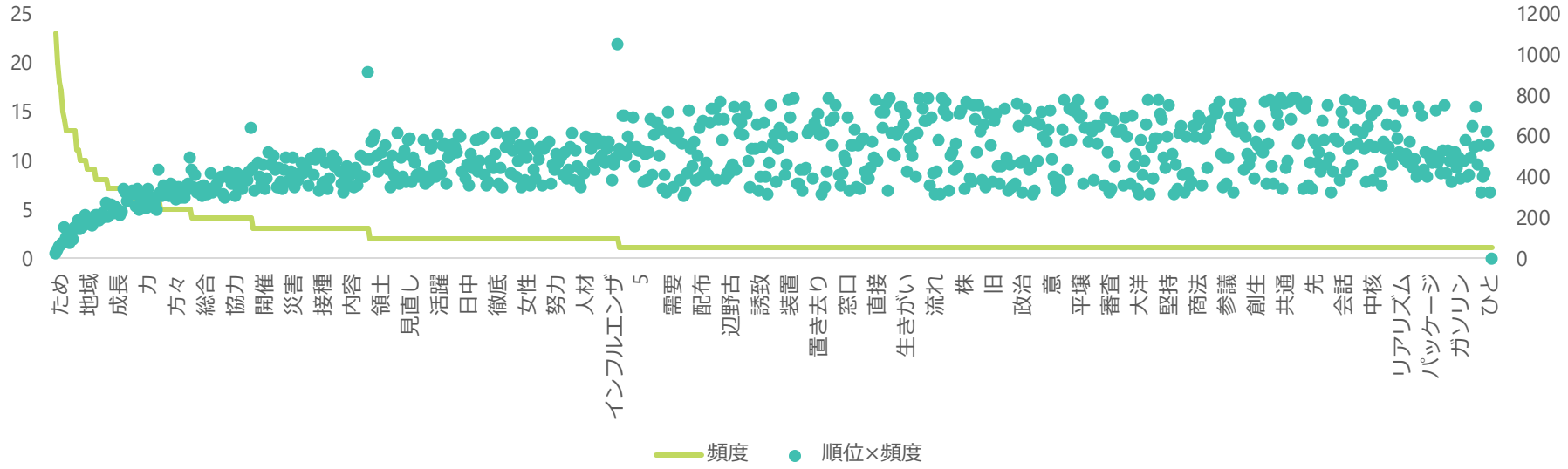
bigram		度数	相対度数
美味しい	寿司	1	0.2
寿司	食べる	1	0.2
新鮮な	寿司	1	0.2
寿司	好き	1	0.2
食べる	新鮮な	1	0.2
	合計	5	1

trigram			度数	相対度数
美味しい	寿司	食べる	1	0.25
寿司	食べる	新鮮な	1	0.25
食べる	新鮮な	寿司	1	0.25
新鮮な	寿司	好き	1	0.25
合計			4	1

テキストにおける集計モデル

- Zipfの法則
  - テキストに表れる要素の頻度を大きい順に並べたとき、その順位と頻度の間には、次の関係が成り立つ。
  - 順位(r)×頻度(f)≒定数(c)

第二百十回国会における岸田内閣総理大臣所信表明演説
https://www.kantei.go.jp/jp/101_kishida/statement/2022/1003shoshinhyomei.html



ジップの法則の拡張

- 順位(r)×頻度(f)≒定数(c)の拡張

$$f_r = \frac{c}{r^a}. \ (r = 1, 2, 3, \dots, n) \quad (1)$$

$$f_r = \frac{c}{(b+r)^a} \quad (r = 1, 2, 3, \dots, n) \quad (2)$$ Zipf-Mandelbrot法則

- a, b, cはパラメータ。(1)の両辺の対数をとると、

$$\log(f_r) = \log(c) - a \log(r)$$

- これは順位(r)と頻度(f)を対数変換した線形回帰式y=α+βx

タイプ・トークン比

- Type-Token Ratio
- テキスト中の語彙の豊富さを示す指標
- 延べ語数Nに対する異なり語数Vの比率で表す。

$$TTR = \frac{V}{N}$$

  - 述べ語数(N)：テキストの中に用いられた総単語数
  - 異なり語数(V)：単語の種類数

首相の所信表明演説の語数とタイプ・トークン比
(安倍、福田：金(2008), 岸田：新たに作成)

	N	V	V/N
安倍	4343	1221	0.2811
福田	3423	928	0.2711
岸田	5142	790	0.1536

TF-IDF

- 語の重みについての指数
- TF：term frequency
  - テキストdにおける語tの頻度(tf)
- IDF：inverted document frequency
  - 語tが検索対象の中のどれくらいのテキストに使用されているかの指標。
  - $IDF = log\left(\frac{N}{df}\right)$　df: 語tを含むテキストの数、N: テキストの総数

	国民	生活	安心	安全
d1	1	0	2	1
d2	3	2	1	0
d3	0	1	0	2
d4	0	3	2	1
d5	0	2	3	0

金(2008), p. 61

$$TF - IDF = tf \times log\left(\frac{N}{df}\right)$$

表の「国民」のTF-IDF = 1×log(5/2)=0.9163

テキストマイニングツール

- AIテキストマイニングツール　https://textmining.userlocal.jp/
  - 許諾なく論文などで無料で使用可。ただし、論文中、脚注や参考文献等に、ツール名とURLを明記。
  - https://textmining.userlocal.jp/questions#use_q1
- jReadability https://jreadability.net/sys/ja
  - 解析結果をcsv形式で保存可能。使用を明記。
- KHCoder https://khcoder.net/
  - 共起ネットワーク、分類分析、クラスター分析、コレスポンデンス分析などの高度な解析機能を備える。

大規模言語モデル（**LLM**）とは？

- **定義**
  大規模言語モデル（Large Language Model, LLM）は、自然言語処理（NLP）に特化したAIモデルで、大量のデータを基に構築され、テキストの生成、要約、翻訳、質問応答、分類など多岐にわたるタスクを遂行。
- **代表例**
  - **GPT-4**: OpenAIが開発。多様な分野の質問に対応し、高い文章生成能力を持つ。
  - **BERT**: Googleが開発。文脈理解を重視し、検索エンジンの最適化にも利用される。
  - LLaMA（Meta開発）やPaLM（Google開発）もLLMの重要な例。

LLMと従来の形態素解析の比較

項目	従来の形態素解析 (MeCab等)	大規模言語モデル (LLM)
データ処理単位	単語・形態素	文脈全体
手動設定の必要性	高い（辞書の設定など）	低い（自己学習型）
新語・未知語対応	制限あり	高い（文脈で推測可能）
応用範囲	分析的（単純な統計分析）	生成的（文章生成、翻訳など）

キーワード検索(Mecab)から意味検索(BERT以降)へ

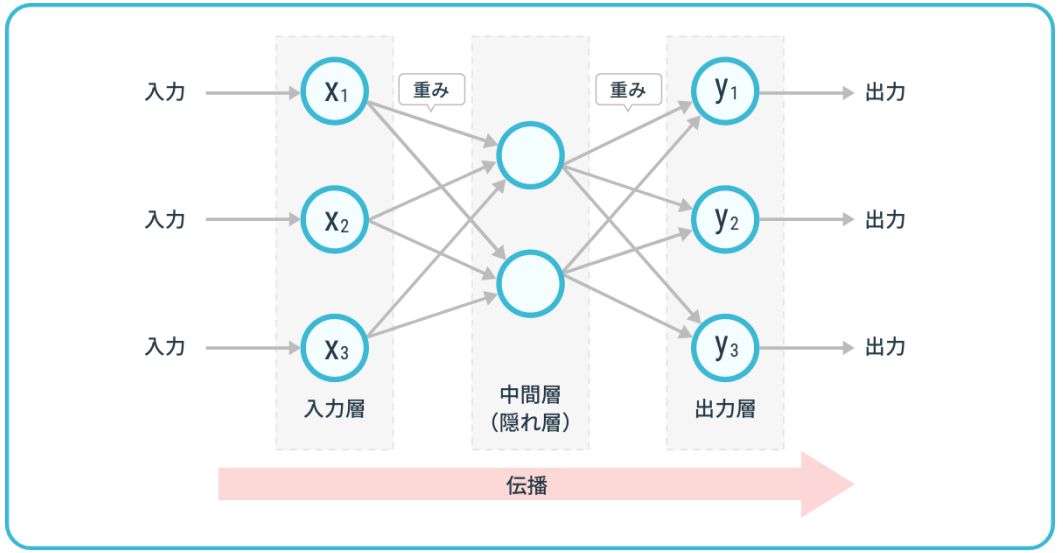
- **(1) 初期検索アルゴリズムとPageRank（1998年開発）**
  - ウェブページ間のリンク構造を評価し、重要なページを特定。
  - ページの関連性をリンクの「信頼性」で測定。
- **(2) クエリ処理の形態素解析（2005年以降）**
  - MeCabや形態素解析が日本語検索で採用され、クエリの正確な処理を実現。
- **(3) 検索の意味的処理（2013年以降）**
  - **Word2Vec**の導入により、単語間の意味的類似性を測定可能に。
- **(4) BERTによる文脈理解（2018年開発、2019年検索に導入）**
- **技術の概要**:
  - 文脈を双方向（前後の単語を参照）で理解するトランスフォーマーベースのモデル。
  - クエリの意図やニュアンスをより正確に理解可能。
- **検索への影響**:
  - 従来のキーワードマッチングから意味ベースの検索に進化。

Word2Vec

- 単語を低次元のベクトルとして表現する手法で、主にGoogleが開発した自然言語処理（NLP）のアルゴリズム。
- ニューラルネットワークを使用して単語間の意味的な類似性を学習する。
- 主なモデル
  - **CBOW（Continuous Bag of Words）**
    - 文脈（周辺単語）から中心単語を予測するモデル。
    - 例えば、"I like ___" という文から "apples" を予測するように訓練。
    - 文脈の単語を平均化し、ターゲット単語を予測する。
  - **Skip-gram**
    - 中心単語から文脈単語を予測するモデル。
    - 例えば、"apples" という単語から、その周囲にある "I" や "like" を予測する。
    - 大きなコーパスで有用で、特に低頻度の単語の学習に強い。

ニューラルネットワーク

- Neural Networkは、脳の神経回路を模倣したアルゴリズムで、データを学習し、パターンを認識するための計算モデル
- ニューラルネットワークは以下の層で構成：
    1. **入力層（Input Layer）**
        1. モデルに入力されるデータを受け取る層。
        2. 例：画像のピクセル値や数値データ。
    2. **隠れ層（Hidden Layers）**
        1. 入力データを処理し、特徴を抽出する層。
        2. 隠れ層の数が多い場合、「深層ニューラルネットワーク（Deep Neural Network, DNN）」と呼ばれる。
    3. **出力層（Output Layer）**
        1. 問題に応じた結果を出力する層。
        2. 例：クラス分類（猫か犬か）、数値予測（家賃の金額）。



https://aismiley.co.jp/ai_news/neural-network/

BERT

- Bidirectional Encoder Representations from Transformers
- Googleが開発した自然言語処理（NLP）のモデルで、従来の手法に比べて文脈の理解が大幅に向上。
- **双方向性（Bidirectional）**文章を左右両方向から同時に解析。
  - 従来のモデル（例: Word2VecやRNN）は片方向性が多く、文脈を一方向にしか考慮しないが、BERTは単語の前後関係を同時に考慮。
- **トランスフォーマー（Transformer）**BERTはトランスフォーマーアーキテクチャのEncoder部分を使用。
  - Transformerは、自己注意（Self-Attention）メカニズムを活用して文脈を効率的にモデル化。

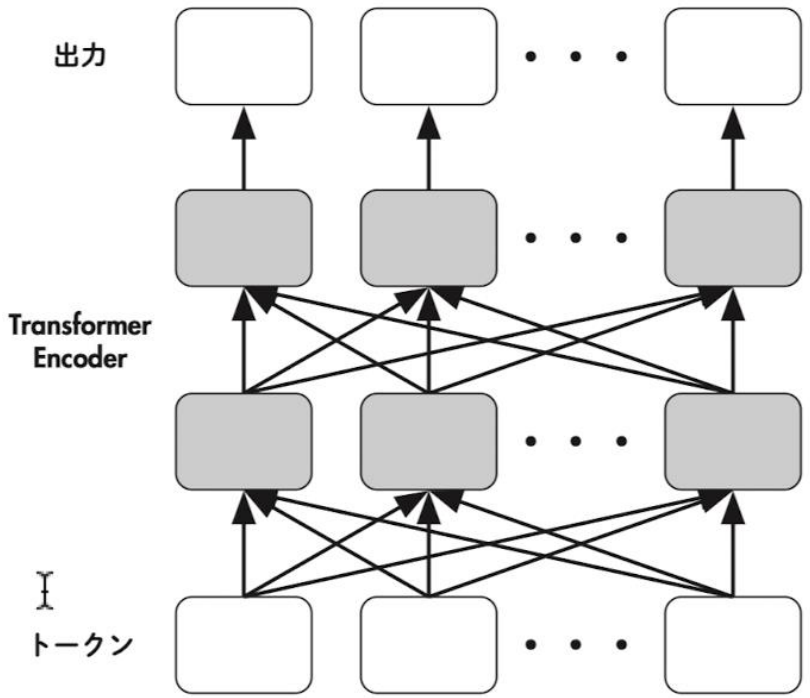


図 3.1　BERT の構造

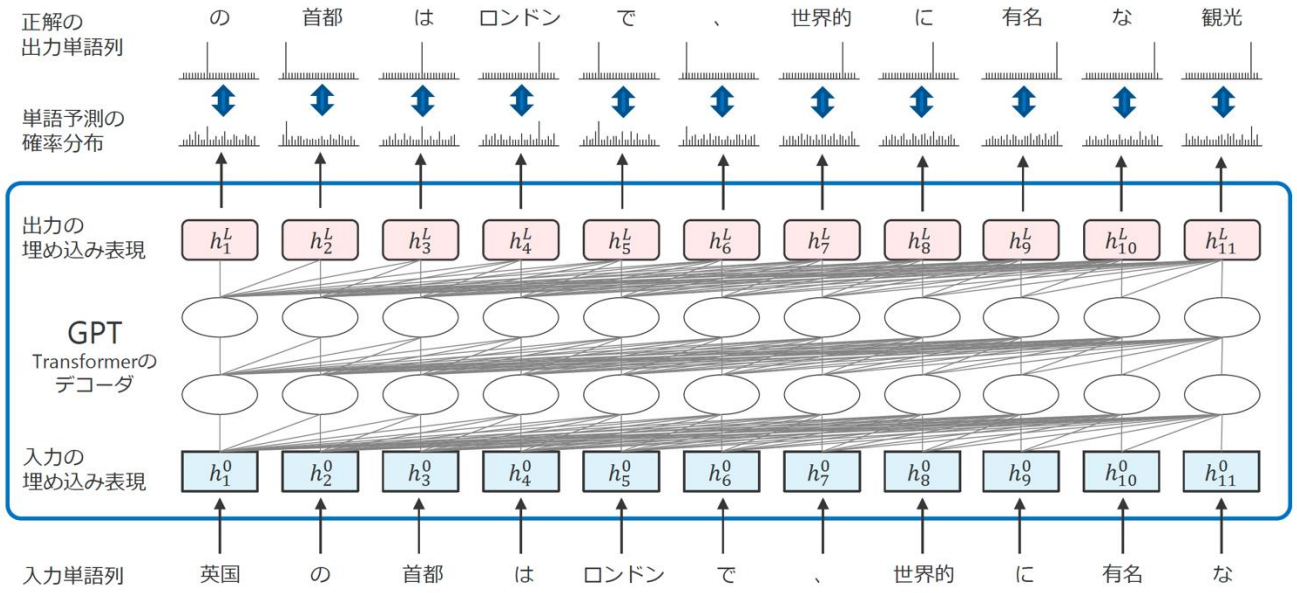
近江崇宏 他 (2021), p. 31より引用

Transformer

- ニューラルネットワークの一つ。
- 自己注意（Self-Attention）メカニズムと並列処理による効率的な学習
- Transformerは以下の2つの主要部分から構成
  - **エンコーダー（Encoder）**
    - 入力文をエンコードして潜在表現（ベクトル）を生成。
  - **デコーダー（Decoder）**
    - エンコーダーの出力を用いて、目標出力文を生成。
  - 通常、エンコーダーとデコーダーは複数層（例えば6層）で構成。

GPT

- Generative Pre-trained Transformer
- Transformerを基盤にした自然言語処理モデルで、Decoder部分を改良して使用
  - **入力埋め込み（Input Embedding）**
    - 入力された単語を固定次元のベクトルに変換。
    - 位置エンコーディング（Positional Encoding）を加算して単語の順序情報を保持。
  - **自己注意（Self-Attention）**
    - 入力文中の単語間の関連性を計算し、どの単語に注意を払うべきかを判断。
    - **マスク付き自己注意（Masked Self-Attention）**を使用：
      - 未来の単語は見えないようにし、生成済みの単語のみを参照。
  - **フィードフォワードネットワーク（Feed-Forward Network）**
    - 各単語の特徴をさらに処理し、次の層に伝達。
  - **出力層**
    - 最終的に、次に生成される単語の確率分布を出力。



https://speakerdeck.com/chokkan/20230327_riken_llm?slide=16から抜粋

「文脈理解」の**BERT**、「テキスト生成」の**GPT**

- **BERT**
  - **構造**: トランスフォーマーの**エンコーダー**部分のみを使用。
  - **双方向性**: 入力文の前後両方の文脈を同時に考慮。
    - 例: "I went to the [MASK] to buy apples." では、「[MASK]」の前後を見て「store」を予測。
- **GPT（Generative Pre-trained Transformer）**
  - **構造**: トランスフォーマーの**デコーダー**部分を改良して使用。
  - **単方向性**: テキストを左から右（順方向）に処理。前の単語だけを使って次の単語を予測。
    - 例: "I went to the store" → 次に「to buy apples」を生成。

LLM進展の背景

- **計算能力の向上**

  モデルサイズは数十億から数兆のパラメータに拡大。クラウドや分散コンピューティングの進展により、これらのモデルが現実的に動作可能になった。

- **トレーニングデータの規模**

  書籍、論文、ウェブデータ、会話記録など、多様なデータセットを使って学習。この膨大なデータがモデルの汎用性を向上させている。

- **アルゴリズムの改良**

  トランスフォーマー構造（Attention Mechanism）を採用し、効率的かつ精度の高い学習が可能になった。

課題と今後の方向性

- **コストとエネルギー効率**
  モデルのトレーニングには膨大な計算資源が必要。環境負荷を軽減する方法の模索が進行中。
- **データバイアス**
  学習データが偏っている場合、応答にも偏りが出る可能性がある。
- **応用範囲の拡大**
  医療、法律、金融などの専門分野向けに、LLMを特化させる動きが加速。

AIを活用したテキストマイニング

1. ソーシャルメディア分析
2. ニュース記事の要約とトレンド抽出
3. 顧客レビューの分析
4. 学術論文の傾向分析
5. 顧客センチメントの地域別解析
6. 医療・健康分野の応用

1. ソーシャルメディア分析

- **テーマ**: X(旧twitter)データを活用した世論分析
- **具体例**:
  - ハッシュタグ（例: #SDGs）や特定のキーワードに関連するツイートを収集。
  - **AIツール**: Hugging FaceのTransformerを用いた感情分析やトピック分類。
  - **結果の可視化**: ツイートのポジティブ・ネガティブ傾向をグラフで表示し、イベントや政策に対する人々の感情変化を追跡。
- **応用**: 政治キャンペーンの効果測定、マーケティング戦略の最適化。

次回

- テキストマイニングに関する演習を行います。

参考文献

- 金明哲(2009)『テキストデータの統計科学入門』,岩波書店, 978-4000057028
- 小林雄一郎(2017)『Rによるやさしいテキストマイニング』, オーム社, 978-4274220234
- 科学雑誌Newton (2020) 『ニュートン式超図解 最強に面白い！！ 人工知能　ディープランニング編』, 株式会社ニュートンプレス, 978-4315521856
- 近江崇宏 他 (2021) 『BERTによる自然言語処理入門 ―Transformersを使った実践プログラミング―』,オーム社, 978-4274227264
- 山田 育矢 他(2023)『大規模言語モデル入門』, 技術評論社, 978-4297136338
- 杉山将 (2024)『教養としての機械学習』,東京大学出版会, 978-4130634595