Analytics Design Lab

人工知能技術コンソーシアム セミナー
【自然言語処理の潮流を一挙マスター！】
AI技術で進化するテキストアナリティクス最前線
<第1部>

いまさら聞けない自然言語処理のAI技術の体系

テキストマイニング・トピックモデル・深層学習モデル・大規模言語モデル

株式会社アナリティクスデザインラボ
代表取締役　野守耕爾

2024年10月23日

第1部の講演内容

第1部では、テキストマイニングからトピックモデル、深層学習モデル、大規模言語モデルといった自然言語処理のAI技術を体系的に解説します

0. 会社紹介

1. テキストマイニング

2. 自然言語処理におけるAI技術の俯瞰

3. 従来の自然言語処理技術

4. トピックモデル

5. 深層学習モデル （第3次AIブーム 前半編）

6. 大規模言語モデル （第3次AIブーム 後半編）

7. 大規模言語モデルとテキストマイニング

8. まとめ

0. 会社紹介

会社紹介と代表紹介

企業様のデータ分析・活用を支援させて頂くコンサルティング会社で、これまでのアカデミックな研究とビジネスコンサルティングの両方の経験を活かして2017年6月に設立しました

会社概要

企業様のデータ分析・活用の支援をさせて頂くコンサルティング会社です

Analytics Design Lab

データというスタートから課題の解決というゴールまでをいかにつなげばよいのか、どのようなデータ処理、分析手法、考察、アクションを検討していけばよいのか、というデータ分析を活用するプロセスを企業様の抱える課題や思惑・事情などに応じてしっかりとデザインし、それを実行することで企業様の課題解決を支援します。

設立	2017年6月1日
事業内容	● 企業におけるデータ活用のコンサルティング ● 新しいデータ分析技術の研究開発
資本金	5,000,000円
所在地	東京都中野区東中野1-58-8-204
URL	http://www.analyticsdlab.co.jp/

代表略歴：野守耕爾

■ 2012年3月
早稲田大学大学院 創造理工学研究科
経営システム工学専攻　博士課程修了
博士（工学）

- 人間行動の計算モデルの開発を研究
（専門領域：人間工学）
- 2010年4月～2012年3月
独立行政法人日本学術振興会 特別研究員に採用

■ 2012年4月～（技術研修生としては2008年～）
独立行政法人産業技術総合研究所
デジタルヒューマン工学研究センター　入所

- センシング技術を応用した子どもの行動計測と人工知能技術を応用した行動の確率モデルの開発を研究

■ 2012年12月～
デロイトトーマツグループ 有限責任監査法人トーマツ
デロイトアナリティクス　入所

- データサイエンティストとしてビッグデータを活用したビジネスコンサルティング及び分析技術の研究開発に従事

■ 2017年6月～
株式会社アナリティクスデザインラボ 設立

コンサルティングサービスの概要

弊社が分析を実施しご提供する「分析受託」、お客様が実施される分析を助言する「アドバイザリー」、弊社実施の分析をお客様にトランスファーする「テーラー研修」がございます

分析受託サービス

お客様のデータをお預かりして弊社がデータ分析を実施し、結果をご報告します

- お客様の業務課題とご提供頂くデータに応じて、弊社がデータ分析の設計を行い、実行します
- 弊社による分析の実施結果をご報告し、その報告書を成果物としてご納品します
- 分析の実施にかかる期間（作業工数）から費用をお見積りします

アドバイザリーサービス

お客様ご自身で実施されるデータ分析・活用のご助言、ご指導をします

- お客様の業務課題の解決に効果的なデータ分析・活用についてご助言します
- お客様が実施される具体的なデータ分析の作業についてもご指導します
- 弊社がご納品する成果物はございません
- １回〇時間の訪問助言を何回ご提供するのかによって費用をお見積りします

テーラー研修サービス

弊社が実施した分析の内容をお客様で実施できるように、その手順を全てレクチャーします

- 「分析受託サービス」で弊社が実施した分析について、実施手順マニュアルや分析のプログラムファイルのご提供とともに解説し、お客様で同様の分析を実行できるように技術トランスファーします
- 「分析受託サービス」の費用に加え、マニュアルの作成や研修の実施などにかかる工数から費用をお見積りします

過去の実績（１）

テキストデータの分析を強みに、Web上の口コミやアンケートの自由記述、コールセンターの問い合わせ履歴、特許文書など、様々なテキストデータの分析を提供してきました

テキストデータを対象とした過去のコンサルティング実績　（デロイトトーマツでの実施案件を含む）

データの種類	クライアント業種	プロジェクト概要	期間
Web口コミ	地方自治体	観光口コミデータを活用した温泉観光地のニーズ分析	2ヶ月
	地方自治体	観光口コミデータを活用した広域観光圏の特徴分析	2ヶ月
	地方自治体	観光口コミデータを活用した観光テーマ抽出と広域ルート検討	4ヶ月
	家電メーカー	製品口コミデータを用いた機能と満足度との関係モデル構築	2ヶ月
	住宅メーカー	戸建て住宅の口コミデータを用いた顧客評価の把握と競合他社分析	3ヶ月
アンケート	地方自治体	市民意識調査の自由記述データを用いた市民ニーズの抽出と効果的な行政施策の検討	2ヶ月
	住宅メーカー	研修受講者アンケートの自由記述データを用いた新規システムのニーズ抽出と改善検討	2ヶ月
	公共事業会社	社内従業員アンケートの自由記述データを用いた従業員のモチベーション傾向分析	1ヶ月
	公益事業会社	消費者アンケートの自由記述データを用いた消費者意見の特徴とその要因関係の分析	2ヶ月
コールセンター	公共事業会社	問い合わせデータを用いた顧客接点業務の課題抽出	2ヶ月
	プリンタメーカー	問い合わせデータを用いた製品の不具合傾向の分析	1ヶ月
	食品メーカー	問い合わせデータを用いた顧客別・商品別の問い合わせ傾向およびニーズの分析	2ヶ月
	ポンプメーカー	ITヘルプデスクの問い合わせデータを用いた質問傾向の分析	1ヶ月
	住宅メーカー	メンテナンス問い合わせデータを用いた不具合問い合わせの類型化と発生傾向の分析	3ヶ月
	ビルメンテナンス会社	メンテナンス問い合わせデータを用いた不具合傾向の分析と業務改善の検討	1ヶ月
特許文書	化学メーカー	特許文書データを用いた技術分類と特許データの整理	1ヶ月
	鉄鋼メーカー	特許文書データを用いた保有技術のターゲット市場探索	3ヶ月
	精密機器メーカー	特許文書データを用いた競合他社の技術動向の把握と自社技術の新規用途探索	3ヶ月
	化学メーカー	国際特許文書データを用いた競合他社の動向把握と用途を実現する重要技術の把握	4ヶ月
	家電メーカー	国際特許文書データを用いた競合会社の技術開発動向の把握と技術戦略の検討	3ヶ月
	家電メーカー	国際特許文書データを用いた競合他社との関係把握と類似特許の探索	2ヶ月

過去の実績（２）

テキストデータに限らずデータ分析全般でサービスを提供しており、また分析の技術的・学術的な観点において学会での受賞実績も複数あります

テキストデータ以外を対象とした過去のコンサルティング実績　（デロイトトーマツでの実施案件を含む）

データの種類	クライアント業種	プロジェクト概要	期間
建築作業記録	住宅メーカー	建築作業の実績データを用いた建築物件の工程日数予測	3ヶ月
機械稼働記録	プリンタメーカー	プリンタの稼働ログデータを用いた故障予測とサービス効率化の検討	2ヶ月
顧客利用情報	保険会社	旅行保険データを用いた事故発生確率および損失額の予測	3ヶ月
顧客利用情報	カード事業会社	ポイントカードデータを用いた顧客セグメンテーションと顧客の来店確率の評価	4ヶ月
アンケート	放送事業会社	アンケートデータを用いた放送局の評価分析	2ヶ月
アンケート	自動車メーカー	ブランド調査データを用いたブランド価値向上の要因構造分析	3ヶ月
ー	システム会社	データサイエンティストの人材育成のための技術指導	6年

学会での受賞歴

受賞年月	学会	受賞内容	発表タイトル
2018年7月	人工知能学会	2018年度全国大会　優秀賞	確率的因果意味解析（PCSA）ーテキストデータを用いたターゲット事象の要因トピックの抽出ー
2018年3月	経営情報学会	2018年春季全国研究発表大会 優秀報告賞	人工知能技術を応用した特許文書分析が生み出す新たな技術戦略の検討
2015年11月	日本マーケティング学会	マーケティングカンファレンス2015 ベストペーパー賞	口コミビッグデータを活用した観光客目線によるテーマ性を持つ広域観光ルートの検討
2015年4月	サービス学会	第2回国内大会 Best Paper Award	観光クチコミデータを用いた類似観光地の発見と満足形成要素の分析
2013年3月	ヒューマンインタフェース学会	学術奨励賞	製品のデザインに関係づけられた乳幼児のよじ登り行動の計算モデル構築と分析
2011年6月	日本人間工学会	大島正光賞（最優秀論文賞）	乳幼児の環境誘発行動を予測する計算モデルの開発

1. テキストマイニング

テキストマイニングとは

テキストマイニングは、テキストデータに記述されている内容の特徴や傾向を把握するための分析手法で、ツールが豊富で使いやすく、ビジネスでもよく活用されています

概要

■ 大量のテキストデータから、その文章に含まれる単語や係り受け表現を抽出し、その出現頻度を集計したり、その出現関係を可視化することで、文章の記述傾向を把握する手法

■ テキストという定性データを統計的に分析可能にする

■ テキストマイニングの2つの基本技術

- **形態素解析**
  文章を意味の持つ最小の言語単位（文字列）に分割し、その文法的素性（品詞など）を付与する
- **構文解析**
  形態素を文節にまとめ，文節間の係り受け関係（主語と述語，修飾語と被修飾語など）を抽出する

函館できれいな夜景を見た

↓ 形態素解析

函館 / で / きれい / な / 夜景 / を / 見 / た

(名詞) (助詞) (形容動詞) (助動詞) (名詞) (助詞) (動詞) (助動詞)

↓ 構文解析

函館で / きれいな / 夜景を / 見た

代表的な公開プログラム

■ 形態素解析のプログラム

- JUMAN （京都大学 黒橋禎夫氏）
- ChaSen （奈良先端科学技術大学院大学 松本裕治氏）
- MeCab （京都大学＆NTTの共同研究 工藤拓氏）

■ 構文解析のプログラム

- KNP （京都大学JUMANをベースに開発）
- CaboCha （工藤拓氏＆松本裕治氏）

代表的なテキストマイニングツール

■ 無償ツール

- KH Coder （立命館大学 樋口耕一氏）
- AIテキストマイニング （ユーザーローカル）

■ 有償ツール

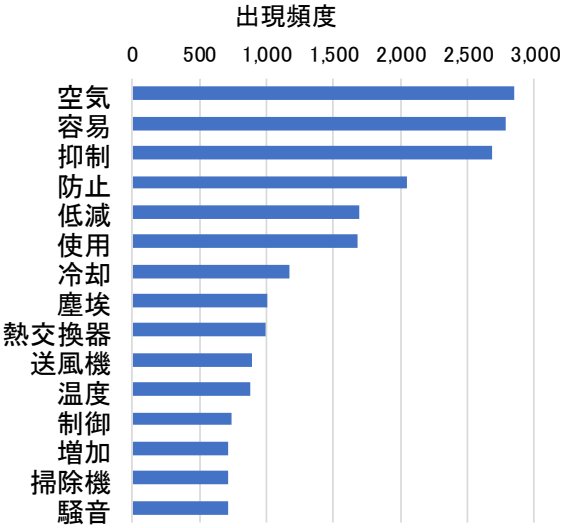
- Text Mining Studio （NTTデータ数理システム）
- 見える化エンジン （プラスアルファ・コンサルティング）
- TRAINA （野村総合研究所）

文章に含まれる単語や係り受け表現をベースとした集計・統計分析を実行することで、文章の特徴を可視化して全体像を把握します

テキストマイニングの可視化例

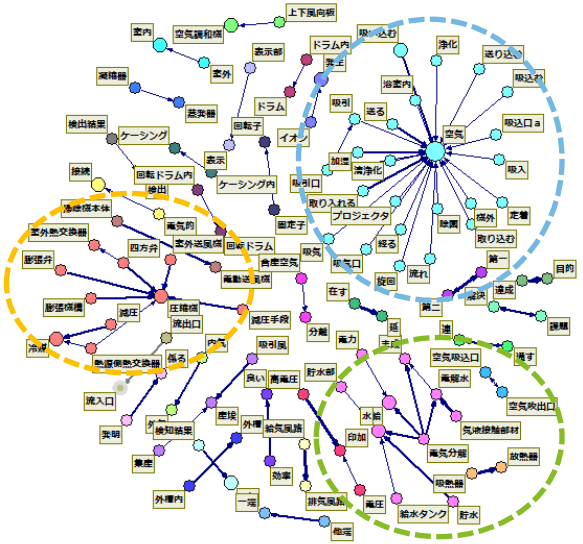
頻度集計

単語や係り受け表現の出現頻度を集計して、どのような記述が多いのか、おおまかな全体像を把握する



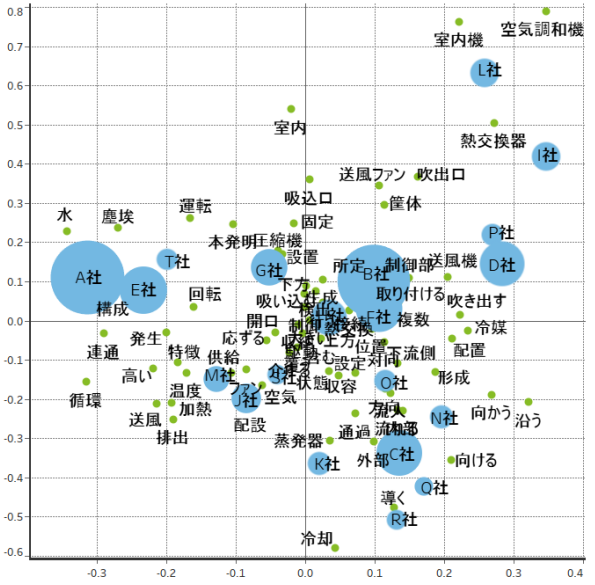
共起ネットワーク

同時に出現しやすい単語同士をネットワークでつなぎ、そのかたまりからどのような話題があるか考察する



コレスポンデンス分析

属性情報と出現単語との対応関係を同じ平面上にマッピングし、その位置関係から属性の傾向を把握する



※特許データをテキストマイニングした例を掲載

テキストマイニングのビジネス活用

例えば、Web上の口コミやアンケートの自由記述、コールセンターの問い合わせ、営業日報、特許文書、有価証券報告書など様々なテキストデータでのビジネス活用があります

テキストマイニングの活用例

Web上の口コミ・SNS

口コミのコメントをテキストマイニングすることで、競合他社と比較して商品やサービスの評判を分析し、現状商品の改善や、新商品の企画に活用したり、SNSのキーワードの頻度の推移から話題のトレンドを把握する

アンケートの自由記述

アンケートの自由記述欄に記載された回答情報をテキストマイニングすることで、選択式回答からは得られない回答者の生の声を把握し、商品企画やマーケティングなどに活用する

コールセンターの問い合わせ

コールセンターの問い合わせ内容をテキストマイニングすることで、顧客のニーズやサービスの改善点を把握したり、頻度の多い問い合わせについてFAQを作成してコール数の削減を図る

営業日報

営業マンの営業記録をテキストマイニングすることで、営業成績別に特徴的なキーワードを比較して成約要因を把握し、営業教育に活用する

特許文書

公開されている出願特許の文書をテキストマイニングすることで、キーワードの推移から技術動向を把握したり、各企業の出願傾向や保有技術の特徴を把握し、研究開発のテーマや、企業間提携・M&Aなどの技術戦略を検討する

有価証券報告書

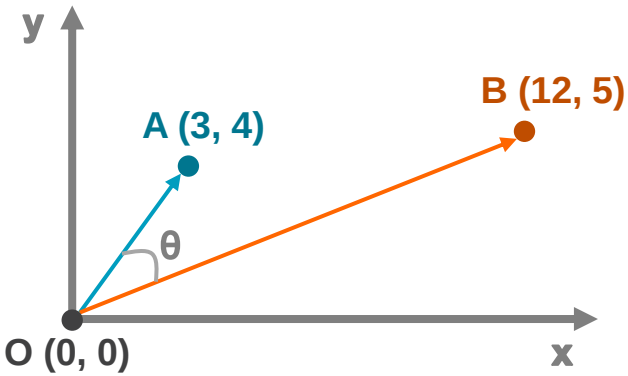
業界における企業の有価証券報告書をテキストマイニングすることで、各社が注力している経営課題の特徴や、そのトレンドを把握し、自社の経営戦略を検討したり、投資先を検討する

2. 自然言語処理におけるAI技術の俯瞰

自然言語処理(NLP: Natural Language Processing)とベクトル表現

自然言語処理とは、人間の言葉を機械で処理することで、そのために文書や単語を数値が並ぶベクトルで表現しますが、AIによって表現力の高いベクトルが獲得可能になりました

x軸とy軸の2次元で表現したベクトル



データ	x	y
A	3	4
B	12	5

■ **AとBの内積（ベクトルのスカラー積）**

$$\vec{A} \cdot \vec{B} = |\vec{A}||\vec{B}|cos\theta$$

$$\vec{A} \cdot \vec{B} = x_a x_b + y_a y_b = 3 * 12 + 4 * 5 = 56$$

$$|\vec{A}| = \sqrt{{x_a}^2 + {y_a}^2} = \sqrt{3^2 + 4^2} = 5$$

$$|\vec{B}| = \sqrt{{x_b}^2 + {y_b}^2} = \sqrt{12^2 + 5^2} = 13$$

■ **AとBのコサイン類似度（ベクトル間の類似度）**

$$\cos\theta = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|}$$

$$cos\theta = \frac{56}{5 * 13} = 0.862 \quad (\theta \approx 31^\circ)$$

単語を軸とした文書のベクトル化

■ 文書に含まれる各単語を軸とし、その出現頻度で文書をベクトル化する（Bag-of-Words）

■ 軸の意味＝単語の意味となり、解釈しやすい

データ		部屋	広い	快適	空調	良い	効く	綺麗	清掃	風呂
A	部屋が広く快適で、部屋の空調も良く効きました。	2	1	1	1	1	1	0	0	0
B	部屋は綺麗に清掃されていて、お風呂も快適でした。	1	0	1	0	0	0	1	1	1

深層学習で獲得する表現力の高いベクトル

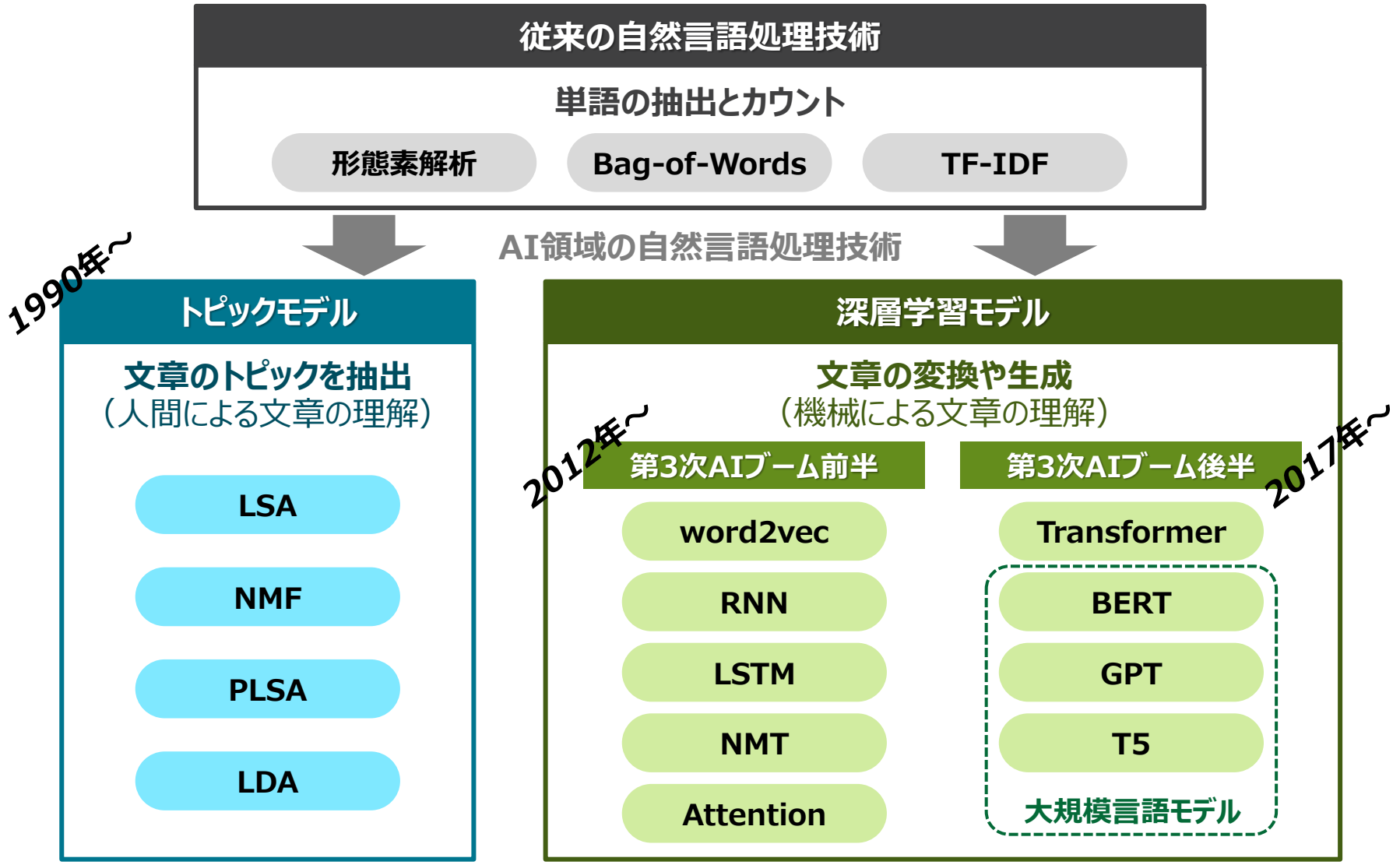
■ 深層学習でベクトルの各数値を最適に推定することでより表現力の高いベクトルを得る

■ 軸の意味も数値の意味も分からない

データ		x1	x2	…	x1000
A	部屋が広く快適で、部屋の空調も良く効きました。	1.846	-0.952	…	0.286
B	部屋は綺麗に清掃されていて、お風呂も快適でした。	0.673	0.438	…	-2.165

自然言語処理の技術整理

自然言語処理技術は、単語の抽出と頻度を集計する形態素解析を従来技術に、文章のトピックを抽出するトピックモデルや、文章の変換や生成をする深層学習モデルがあります



3. 従来の自然言語処理技術

Bag-of-Words (BoW)

Bag-of-Wordsは各文書の単語の出現頻度をカウントしたデータで、最もシンプルに文書をベクトル化できる手法ですが、これだけでもテキストデータの様々な定量分析が可能です

概要

- 各文書にどの単語が何回出現したのかをカウントする方法で、出現頻度という数値によって文書一つ一つをベクトルとして表現したもの
- 最もシンプルな文書のベクトル表現だが、これだけでもテキストデータの様々な定量分析が可能で、テキストマイニングのツールで実行される可視化や統計解析は基本的にこのデータ形式をベースにしている

課題

- 全ての単語は同等に扱われ、その単語が全文書で見たときにどれくらい重要であるかは分からない
- 単語の位置や順序、使われ方、文脈上の意味といった情報は保持しておらず、その文字列の出現頻度のみの情報である
- 単語の種類の数がベクトルの次元数となるため、高次元のデータとなり複雑で、計算処理が難しくなる

イメージ図

文書ID	ホテルの口コミコメント	部屋	広い	快適	空調	良い	効く	綺麗	清掃	風呂	駅	近い	便利	人	対応	丁寧	朝食	美味しい	レストラン
1	部屋が広く快適で、部屋の空調も良く効きました。	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	部屋は綺麗に清掃されていて、お風呂も快適でした。	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
3	駅の近くで便利で良く、部屋も広くて良かったです。	1	1	0	0	2	0	0	0	0	1	1	1	0	0	0	0	0	0
4	人の対応が良く、部屋も丁寧に清掃されていました。	1	0	0	0	1	0	0	1	0	0	0	0	1	1	1	0	0	0
5	朝食が美味しく、レストランも広くて綺麗で良かったです。	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1

TF-IDFはBag-of-Wordsで同等に扱われていた各単語の重要度を数値化する前処理であり、文書の特徴をより捉えた分析が可能となります

概要

- 各文書における単語の重要度を数値化する処理であり、TFとIDFを掛け合わせた値を取る
- TF (Term Frequency)
  - 各文書の中における単語の頻度を示す値
  - 頻度が多い単語ほど重要である
  - 文書の長さの影響を受けるため、各文書の全単語頻度で除すことが多い
- IDF (Inverse Document Frequency)
  - ある文書で頻出する単語でも、それがどの文書でも頻出するなら重要とはいえない
  - IDF=log（全文書数 / 単語wが出現する文書数）
  - 多くの文書で現れる単語は値が小さく、特定の文書にしか現れない単語は値が大きくなる
- TF-IDFは、テキストデータの分析でよく用いられる前処理であり、例えば文書間の類似度の分析では、TF-IDFの処理をしたベクトルを使い、そのcos類似度を計算することが多い

イメージ図

Bag-of-Words

ID	部屋	広い	快適	空調	良い	合計頻度
1	2	1	1	1	1	6
2	1	0	1	0	0	2
3	1	1	0	0	2	4
4	1	0	0	0	1	2

TF-IDF

ID	部屋	広い	快適	空調	良い
1	0	0.116	0.116	0.231	0.048
2	0	0	0.347	0	0
3	0	0.173	0	0	0.144
4	0	0	0	0	0.144

（例）ID=3の「良い」のTF-IDF
TF = 2 / 4 = 0.5
IDF = log (4 / 3) = 0.288
TF-IDF = 0.5 * 0.288 = 0.144

4. トピックモデル

LSA (Latent Semantic Analysis)

LSAは「文書×単語」の行列を特異値分解によって「文書×トピック」「トピック×トピック」「トピック×単語」に分解することでトピックを抽出しますが、結果に負の値を含みます

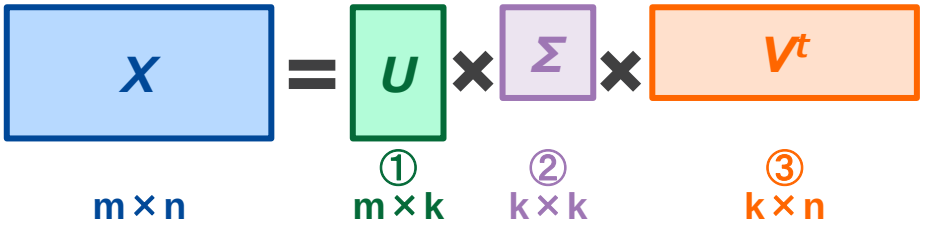
概要

- LSA（潜在意味解析）は、「文書×単語」の行列（Bag-of-Words）を特異値分解によって次元削減することでトピックを抽出する手法で、1990年に発表された
  - 「文書×単語」行列を以下3つの行列に分解する
    ①文書×トピック （左特異ベクトル）
    ②トピック×トピック （特異値）
    ③トピック×単語 （右特異ベクトル）
  - 元の行列と分解後の行列の誤差を最小化する
  - 特異値は対角行列であり、左特異ベクトルと右特異ベクトルの行列は各トピック間で直交している（トピックの軸は互いに独立している）
- 大きな値に引っ張られてトピックが抽出される傾向があるため、Bag-of-WordsにTF-IDFなどで重み付けされた行列を用いられることが多い
- 最適解が数学的に保証されており、計算効率も高い
- 分解する行列の要素に負の値を許容しているため、結果の解釈が難しくなる
- 結果が学習データに完全に依存するため、過学習を起こしやすく、新しい文書のトピックは推定できない

イメージ図

LSAの特異値分解

$$X = U\Sigma V^t$$



LSAの結果例

①文書×トピック

U	トピック1	トピック2	トピック3	トピック4
文書1	-0.47	-0.75	-0.46	0.11
文書2	-0.61	-0.10	0.69	-0.37
文書3	-0.27	0.41	-0.54	-0.68
文書4	-0.58	0.52	-0.10	0.62

②トピック×トピック

Σ	トピック1	トピック2	トピック3	トピック4
トピック1	7.7	0	0	0
トピック2	0	2.8	0	0
トピック3	0	0	2.0	0
トピック4	0	0	0	0.6

③トピック×単語

Vt	単語1	単語2	単語3	単語4	単語5
トピック1	-0.51	-0.49	-0.38	-0.37	-0.47
トピック2	-0.27	0.14	0.80	-0.50	-0.09
トピック3	0.47	0.43	-0.35	-0.67	-0.14
トピック4	0.61	-0.74	0.23	-0.18	0.07

NMF (Non-negative Matrix Factorization)

NMFは「文書×単語」の行列を2つの非負の行列「文書×トピック」「トピック×単語」に分解することでトピックを抽出し、結果が非負となるためトピックの解釈がしやすいです

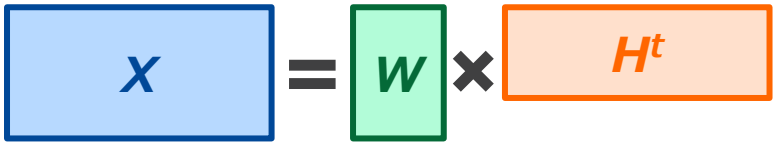
概要

- NMF（非負行列因子分解）は、「文書×単語」の行列（Bag-of-Words）を2つの非負の行列「文書×トピック」「トピック×単語」に分解することでトピックを抽出する手法で、1999年に発表された
- 計算アルゴリズムは、元の行列と分解後の行列の積との誤差を最小化することを目的関数に、初期値を与えた反復計算により最適解を得る
  - 誤差の定義の仕方は、平方ユークリッド距離やKLダイバージェンスなどが使われる
- LSAと比較して、分解後の行列の要素が全て非負であるため、結果の解釈がしやすい
- 分解された「文書×トピック」「トピック×単語」の行列は、各トピックのベクトル間で直交しておらず、各トピックは互いに独立していないため、抽出されたトピックの一部は意味が重複する可能性がある
- 結果が学習データに完全に依存するため、過学習を起こしやすく、新しい文書のトピックは推定できない

イメージ図

NMFの行列分解

$$X = WH^t$$



X	W	Ht
m×n	m×k	k×n
文書×単語	文書×トピック	トピック×単語

NMFの結果例

文書×トピック

W	トピック1	トピック2	トピック3	トピック4
文書1	0.22	0.48	0.16	0.39
文書2	0.35	0.12	0.07	0.88
文書3	0.11	0.58	0.24	0.14
文書4	0.29	0.27	0.60	0.20

トピック×単語

H^t	単語1	単語2	単語3	単語4	単語5
トピック1	4.34	0.03	1.21	2.24	1.87
トピック2	3.41	1.89	2.81	0.46	1.03
トピック3	0.38	2.24	1.09	5.62	0.06
トピック4	1.85	3.66	0.08	2.11	4.16

PLSA (Probabilistic Latent Semantic Analysis)

PLSAはLSAを確率的に発展させた手法で、「文書 × 単語」の同時確率P(文書, 単語)をP(文書 | トピック)、P(単語 | トピック)、P(トピック)に分解することでトピックを抽出します

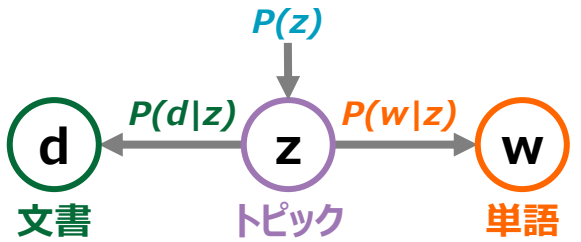
概要

- PLSA（確率的潜在意味解析）は、LSAの処理を確率的な枠組みで発展させたもので、「文書 × 単語」の行列（Bag-of-Words）を確率的に分解することでトピックを抽出する手法で、1999年に発表された
- 同時確率P(文書, 単語)を以下の3つの確率分布に分解して表現し、「文書 × 単語」データから推定する
  ①P(文書|トピック), ②P(単語|トピック), ③P(トピック)
  - ➢ 対数尤度関数を最大化するEMアルゴリズムを実行し、初期値を与えた反復計算により最適解を得る
- 各トピックは互いに独立の仮定を置いている
- LSAでは「文書 × 単語」の行列に対して事前にTF-IDFなどで重みづけする必要があるが、PLSAでは確率的な処理によりそうした重みづけは不要となる
- 文書および単語のトピックに対する関連度が所属確率として出力されるため、結果が解釈しやすい
- 結果が観測データに完全に依存するため、過学習を起こしやすく、新しい文書のトピックは推定できない
  - ➢ 一方で、観測データの再現度が高く、個別性の強い特徴を反映できるモデルと捉えることもできる

イメージ図

PLSAの確率モデル

$$P(d,w) = \Sigma_z P(d|z)P(w|z)P(z)$$



PLSAの結果例

P(文書d | トピックz)

P(d|z)	トピック1	トピック2	トピック3	トピック4
文書1	0.41	0.16	0.06	0.27
文書2	0.29	0.54	0.11	0.06
文書3	0.22	0.13	0.04	0.58
文書4	0.08	0.17	0.79	0.09

P(単語w | トピックz)

P(w|z)	トピック1	トピック2	トピック3	トピック4
単語1	0.11	0.09	0.44	0.20
単語2	0.09	0.38	0.04	0.18
単語3	0.50	0.11	0.29	0.09
単語4	0.08	0.14	0.17	0.31
単語5	0.22	0.28	0.06	0.22

P(トピックz)

P(z)	トピック1	トピック2	トピック3	トピック4
	0.31	0.27	0.23	0.19

$\Sigma_d P(d|z) = 1$

$\Sigma_w P(w|z) = 1$

$\Sigma_z P(z) = 1$

LDA (Latent Dirichlet Allocation)

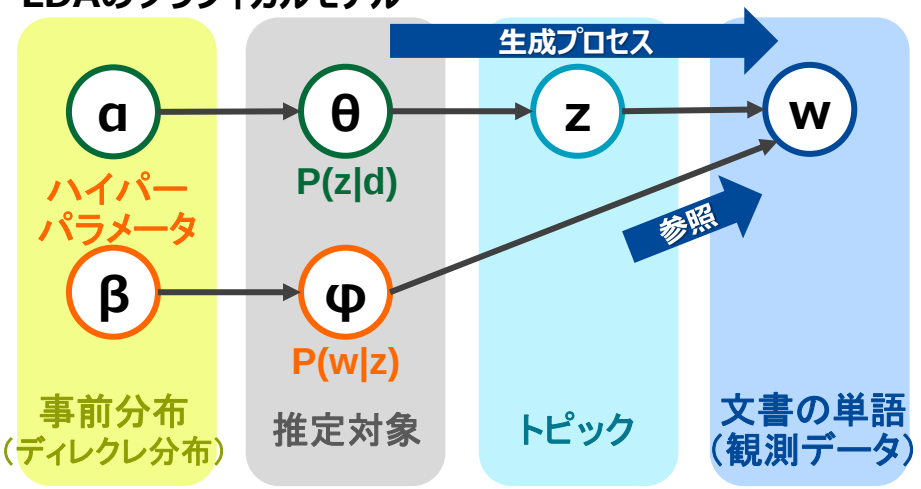
LDAはPLSAをベイズ的に拡張させてディレクレ分布を事前分布に導入した手法で、過学習を抑制でき、新しい文書のトピックも推定できる高い汎化性能があります

概要

- LDA（潜在ディリクレ配分法）は、PLSAをベイズ的に拡張させた手法であり、ディレクレ分布を事前分布に導入しており、2003年に発表され、トピックモデルの中で最もよく使われている手法である
- 推定対象はθ=P(トピック|文書)とφ=P(単語|トピック)であり、推定アルゴリズムは、ギブスサンプリングや変分ベイズ法などが適用され、反復計算される
- PLSAはパラメータは固定的で、観測データのみから直接推定するため、結果が完全に観測データに依存するが、LDAはパラメータは事前分布に従い変動する確率分布とし、観測データと事前分布から推定する
  - ➢ 事前分布を導入することで、確率的なスムージング効果があり、過学習を抑制できる
- LDAは新しい文書についても推定ができ、新しい文書に対応する"θ"を未知とし、"w"（文書に含まれる単語）と学習済みの"φ"と"α"から"θ" を推定する
- ハイパーパラメータαとβによって結果が変動しやすく、この値の設定の仕方、推定の仕方が難しい
- 新しい文書の推定ができる高い汎化性能を持つが、トピックの結果が一般的で抽象度が高くなることがある

イメージ図

LDAのグラフィカルモデル



LDAは、文書d内の各単語wが特定のトピックzから生成されたと仮定する。このトピックzの分布は文書d毎に異なり、その確率分布P(z|d)はθで表す。特定のトピックzが各単語wを生成する確率分布P(w|z)はφで表し、そのトピックzの下で単語wを生成するプロセスではφが参照される。なお、θとφはそれぞれハイパーパラメータαとβを持つディレクレ分布に従う。LDAではこれらの生成過程を逆にたどるアルゴリズムにより、単語w（観測データ）からθとφを推定する。

ハイパーパラメータα,βの大きさとディレクレ分布の特徴

α,β>1	確率が均一化し、トピックは多様な単語に分散する
α,β=1	分布は一様分布となり、PLSAと近い振る舞いをする
1>α,β>0	確率が局所的となり、トピックは一部の単語に偏る

5. 深層学習モデル （第3次AIブーム 前半編）

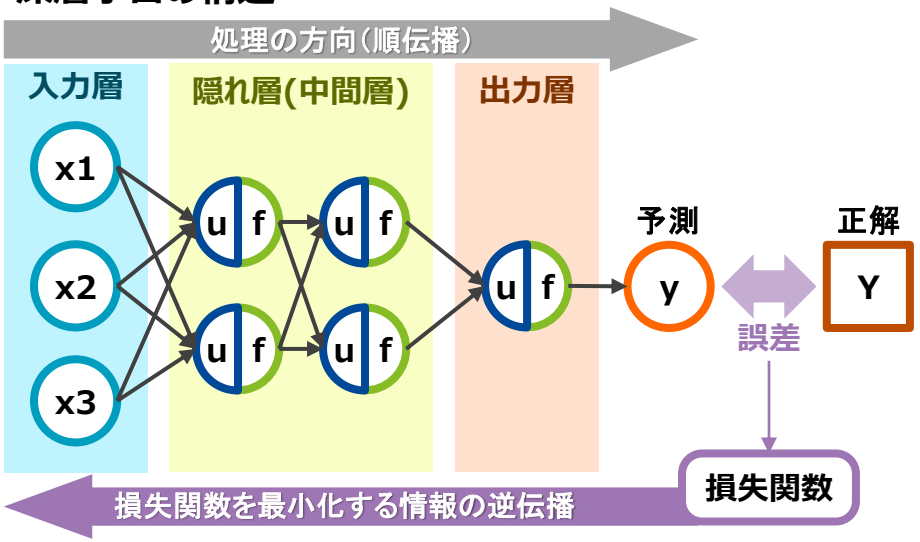
深層学習は多層のニューラルネットワークで、各層は線形変換と非線形変換で複雑な表現力を持ち、線形変換の各重みは誤差逆伝播法で予測誤差を最小にするよう更新されます

概要

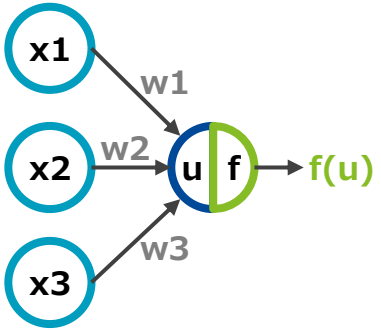
- 深層学習は隠れ層を多層構造にしたニューラルネットワークで、高度なパターン認識や予測を行う
- ニューラルネットワークは入力層、隠れ層（中間層）、出力層で構成され、前の層からの情報を線形変換と非線形変換をして次の層に引き継ぐ
  - 線形変換とは前の層からの情報に重みをかけてバイアスを加えた加重和の計算である
  - 非線形変換とはシグモイド関数やハイパボリックタンジェント関数などの活性化関数を適用する
  - この処理の組み合わせにより複雑な表現力を得る
- 各層の重みとバイアスは、予測誤差を表す損失関数を最小にするように、誤差逆伝播法により更新される
  - 誤差逆伝播法では、出力層から入力層に向かって各層の重みにおける損失関数の勾配（偏微分）を逐次的に計算し、損失関数が最も降下する方向に向けて重みを更新していく（勾配降下法）
- 2006年にトロント大学のジェフリー・ヒントン氏らが深層学習を実現する重要な論文を発表して以降、飛躍的な進歩を遂げ、第3次AIブームが幕を開けた

イメージ図

深層学習の構造



ニューロンの構造



U:入力の加重和（線形変換）

$u = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + b$

w:重み、b:バイアス項

f:活性化関数（非線形変換）

（例）シグモイド関数 $f(u) = \frac{1}{1+e^{-u}}$

他にもtanh関数、ReLu関数など

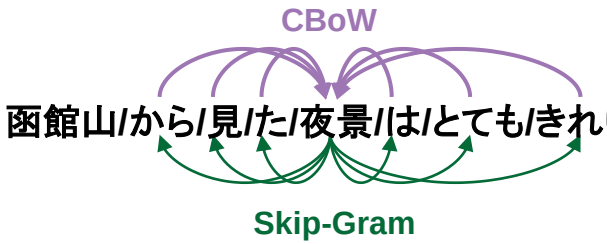
word2vecは大量のコーパスの穴埋め問題を学習したモデルで、単語の意味や類似性を捉えたベクトル表現を得ることができ、単語のベクトル間の演算もできます

概要

- word2vecは2013年に発表された単語のベクトル表現（分散表現）を得る手法であり、これは単語の意味や類似性を捉えており、単語同士の演算もできる
- word2vecは大量のテキストデータ（コーパス）を用いた深層学習モデルで、CBoWとSkip-Gramの2つのアルゴリズムがある
  - CBoW (Continuous Bag of Words)は、ある単語をその周辺単語から予測し、Skip-Gramはある単語からその周辺単語を予測する（穴埋め問題）
- 事前学習されたモデルを使うことで、単語のベクトル表現を容易に得ることができる
  - 例えば「GoogleNews-vectors-negative300」は約300万語に対して300次元のベクトルを得られる
- 各文章に対して単語と同様に「文章ID」の要素を持たせてword2vecの学習を適用することで、単語と同様に「文章ID」のベクトル表現を得ることができ、この手法をdoc2vecという
- word2vecは単語の順序情報が欠落しており、同じ単語でも、文脈で異なる意味を持つ場合は区別できない

イメージ図

CBoWとSkip-Gram



※ウインドウサイズ（周辺単語の数）=3のイメージ

単語の演算

「王」－「男」＋「女」≒「女王」

（例）word2vecで得られる単語のベクトル表現
（4次元のイメージ）

王 = (1.2, 0.6, -0.8, 2.0)
男 = (1.3, 0.5, -1.0, 1.9)
女 = (1.1, 0.7, -0.9, 2.1)
女王 = (1.0, 0.8, -0.7, 2.2)

word2vecは単語を線形性を持つベクトル空間に配置するため単語間の演算が成立する

RNN (Recurrent Neural Network)

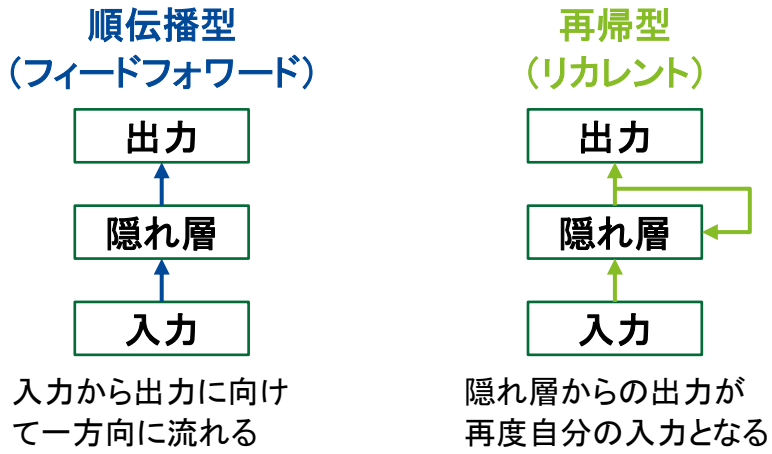
RNNは系列データの処理モデルで、過去の情報を保持して現在の入力を処理することで、単語の順序を考慮できますが、長い系列は過去の情報を現在に反映することが困難です

概要

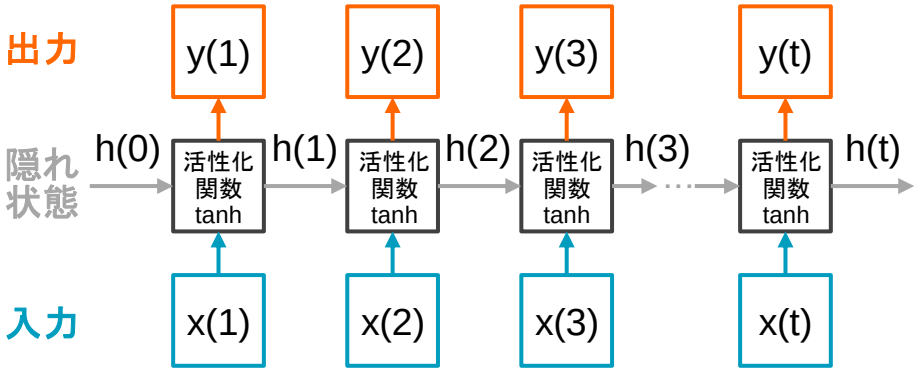
- RNNは文章や時系列情報など、系列データを処理するモデルで、考え方の起源は1980年代に発表された
  - 文章を単語の系列データとして捉え、単語を一つずつ順番に逐次的に処理をする
  - 各ステップで生成した隠れ状態ベクトルが次の隠れ状態の入力にもなる再帰的な処理をする
- 過去の処理情報を保持して、それが現在の入力に影響を与える構造により、単語の順序（文脈）を考慮した処理ができる
- 長い系列データの場合、過去の情報を反映させるのが困難となる「長期依存性の問題」がある
  - 再帰的処理により隠れ状態が次の隠れ状態に連携し、長い系列でネットワークが深くなると、誤差逆伝播法において勾配爆発や勾配消失が起き、過去の遠い情報を最適に保持できなくなる
  - 勾配爆発は、再帰処理により、同じ重みが繰り返し乗算され、勾配が指数関数的に増加する
  - 勾配消失は、再帰処理により、微分が0に近い活性化関数を繰り返し通過し、勾配が急速に小さくなる

イメージ図

ニューラルネットワークの構造の種類



RNNの処理



※h(0)は初期化された隠れ状態（要素ゼロ）

LSTM (Long Short-Term Memory)

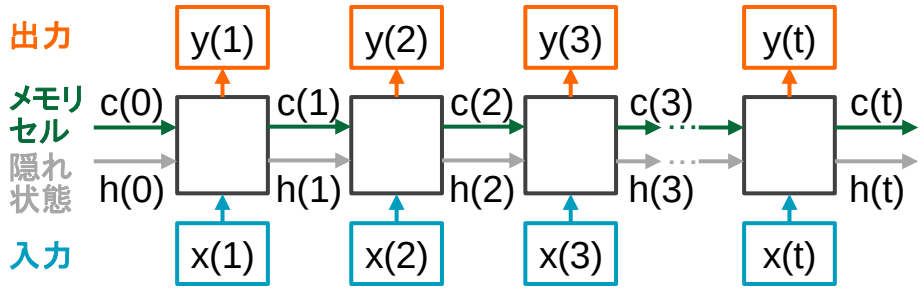
LSTMはRNNの長期依存性の問題を解決したネットワーク構造を持ち、過去の重要な情報を保持し、不要な情報を忘れる能力を有し、長い系列のデータの処理ができます

概要

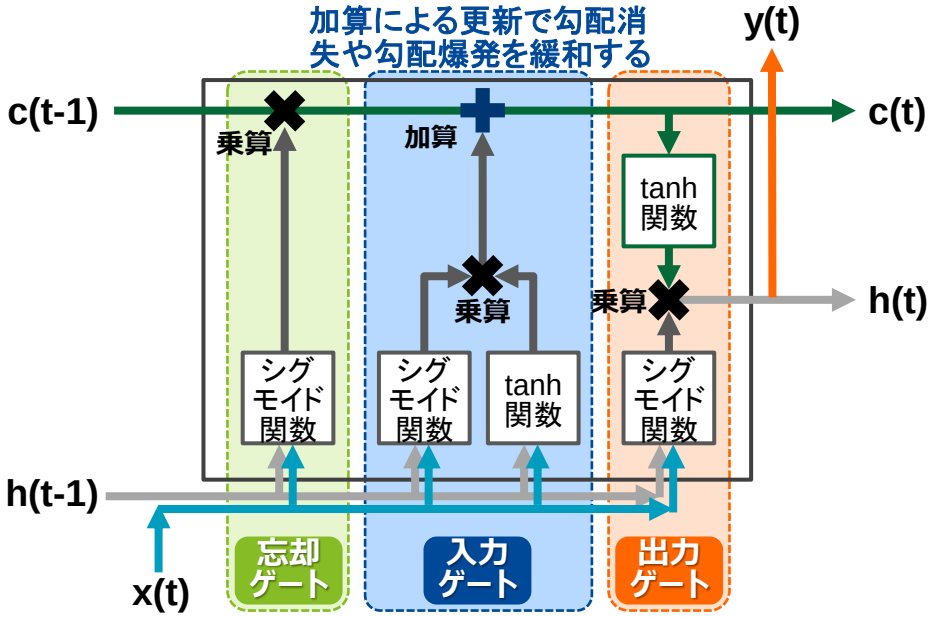
- LSTMはRNNと同じく系列データの処理モデルで1997年に発表されており、長い系列の処理ができる
  - RNNの隠れ状態h（短期記憶）に加え、メモリセルc（長期記憶）の情報がセルからセルに受け継がれる
- 忘却ゲート、入力ゲート、出力ゲートという3つのゲート構造を持ち、各ゲートが情報の流れを制御し、長期記憶が加算によって更新されることで（RNNの更新は乗算のみ）、長期依存性の問題を解決する
  - 忘却ゲートは、受け継がれたメモリセルcの長期記憶の情報のうちどれを消去するか制御する
  - 入力ゲートは、現在の入力xと前の隠れ状態hを使って、新しく記憶するべき情報の候補を生成し、どの情報を入力し記憶させするか制御する
  - 出力ゲートは、次の隠れ状態hと出力yの情報を制御する
- LSTMは勾配消失や勾配爆発の問題を緩和し、RNNよりも長い系列が扱えるが、完全ではなく、RNNが10語程度に対してLSTMでも20語程度と言われている
- LSTMは計算コストが高いという課題もある

イメージ図

LSTMの処理



LSTMのセルの内部



seq2seqは機械翻訳など文章を文章に変換するモデルで、RNNやLSTMで構成されたエンコーダ（文章のベクトル変換）とデコーダ（ベクトルの文章変換）を連結しています

概要

■ seq2seqは系列データを系列データに変換するモデルで、Encoder-Decoderモデルとも呼ばれ、機械翻訳を主な用途とし、2014年に発表された

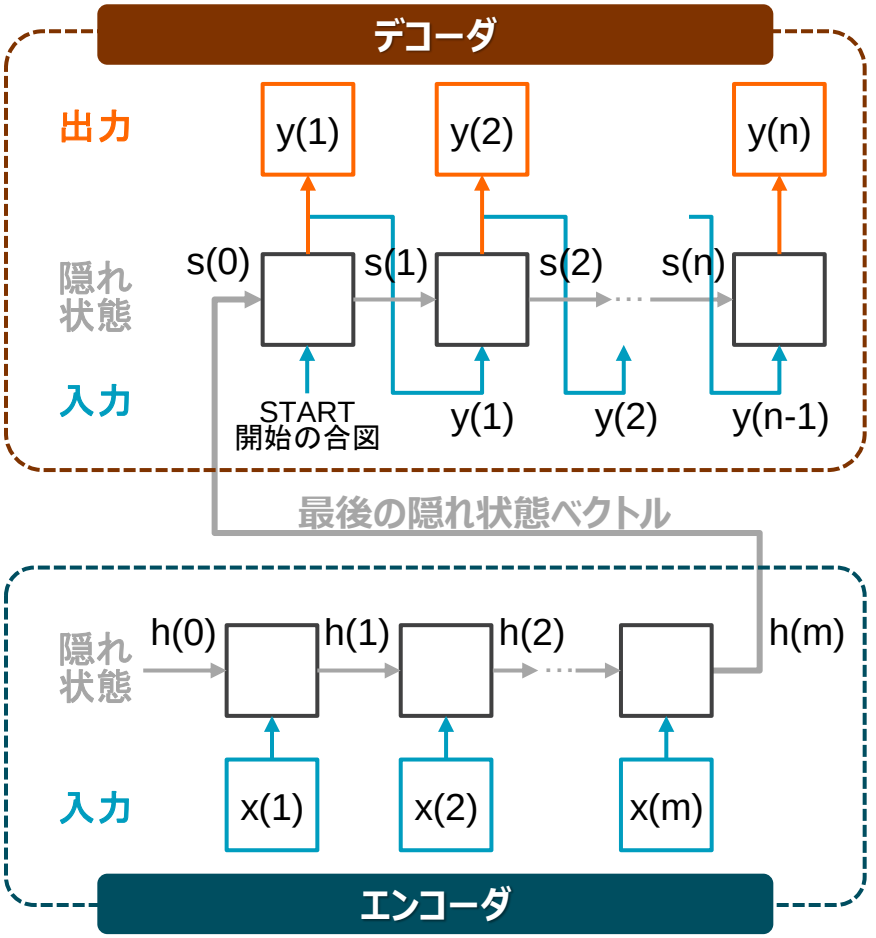
- 文章をベクトルに変換するエンコーダ（seq2vec）と、ベクトルを文章に変換するデコーダ（vec2seq）が連結した構造を持つ

■ RNNやLSTMは単体では、①seq2vec、②vec2seq、③入力・出力の系列数が一致するseq2seqは処理できるが、系列数の一致しないseq2seqは処理できないため、seq2vecとvec2seqを連結して対応したモデル

- seq2vecは、文章を一つのベクトルに変換する処理で、RNNやLSTMの最後の隠れ状態に該当する
- veq2seqは、一つのベクトルを文章に変換する処理で、RNNやLSTMでは、一つのベクトルを入力として単語を生成し、その単語と前の隠れ状態を新たな入力として次の単語を生成し、文章を生成する
- RNNやLSTMは系列ごとに出力を生成するため、単語の品詞割当など入力と出力の系列数が一致するseq2seqは処理できるが、機械翻訳など入力と出力の系列数が一致しないと処理できない

イメージ図

seq2seqの構造（RNNを用いた場合）



※s(0)はエンコーダの最後の隠れ状態ベクトルh(m)となる

Attentionは入力文を出力文に変換する際に、入力のどの単語に注目すべきかを考慮する仕組みで、入力と出力の依存関係を捉えることができ、精度の高い結果を生成できます

概要

■ Attentionは、文章の中でどの単語に注目すべきかを判断する仕組みであり、2014年に発表された

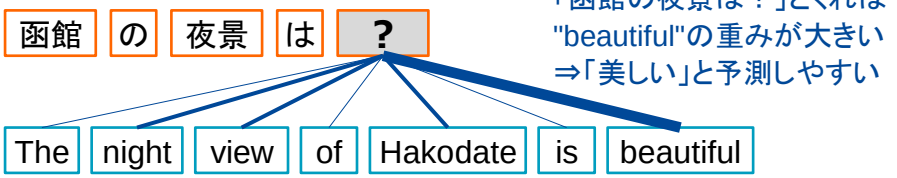
- ➢ 従来のseq2seqは、エンコーダの最後の隠れ状態だけがデコーダに連携されるため、受け渡しできる情報量に限界があり、長い系列ほど精度が落ちた
- ➢ Attentionでは、入力文章を出力文章に変換する際に、入力のどの単語に注目すべきか、あるいは無視すべきかを考慮できるため、出力精度が上がる
- ➢ 遠い所にある単語も含め、入力文章に含まれる全ての単語を参照できるため、RNNやLSTMで課題となっていた長期依存性の問題を補完できる

■ Attentionでは、デコーダの今の隠れ状態に対するエンコーダの各隠れ状態の注目度（Attention weight）を計算し、その重み付きの隠れ状態を足し合わせて文脈ベクトルを作成し、これをデコーダに受け渡す

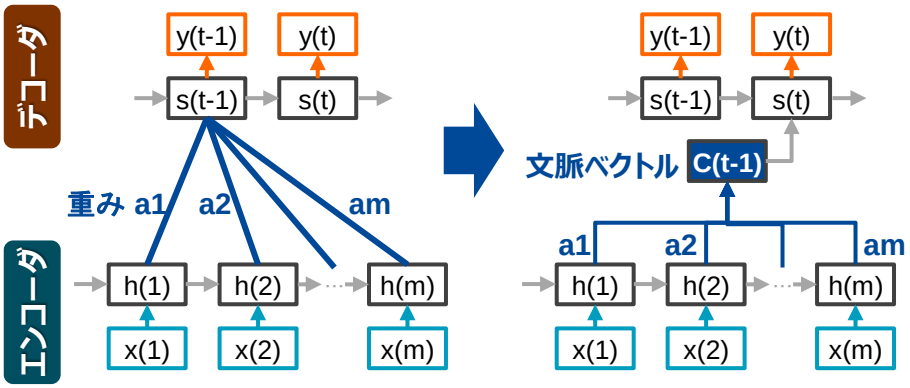
- ➢ この処理をデコーダが新しい単語を生成する度に行い、新たな文脈ベクトルが作成され受け渡される
- ➢ 注目度はデコーダとエンコーダの隠れ状態のペアを入力とする単純なニューラルネットで計算される

イメージ図

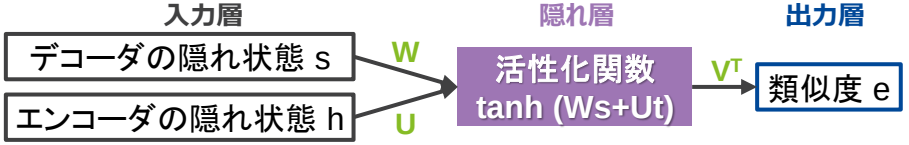
Attentionによる翻訳のイメージ



Attentionの重みと文脈ベクトルの作成



Attentionの重みaを計算するニューラルネットワーク



デコーダの隠れ状態とエンコーダの隠れ状態のペアを入力に、その類似度eを出力としたニューラルネットを学習してそれぞれの重みを最適化し、各類似度eにsoftmax関数を適用して合計が1となる重みaを計算する

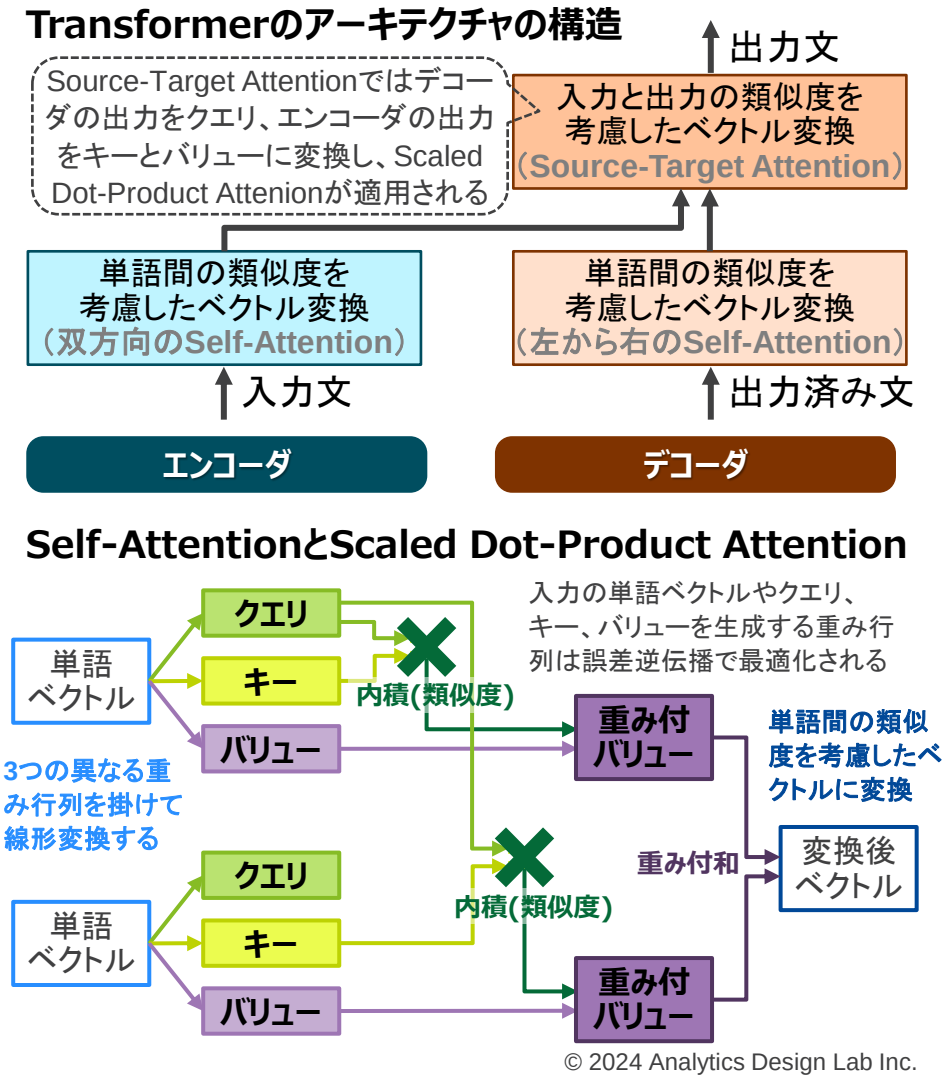
6. 大規模言語モデル （第3次AIブーム 後半編）

TransformerはAttentionだけを用いたエンコーダ-デコーダ構造の深層学習モデルで、高精度かつ高速化を実現し、自然言語処理の分野で大きなブレイクスルーとなりました

概要

- Transformerは、2017年に"Attention is All You Need"というタイトルで発表された、Attentionだけを用いたエンコーダ-デコーダの構造の深層学習モデル
- 3つのAttentionで、表現力の高いベクトルを獲得する
  - Self-Attention
    元々のAttentionは、入力文（Source）の単語と出力文（Target）の単語を対応づけたSource-Target型のAttentionだが、Self-Attentionは、同一文章の中の各単語が他の単語とどの程度関係しているのかを評価し、単語間の依存関係を学習する
  - Scaled Dot-Product Attention
    Self-Attentionにおいて、内積の類似度計算で他の単語との関連性を考慮した単語ベクトルを獲得する
  - Multi-Head Attention
    Self-Attentionを異なる表現空間（ヘッド）で複数パターン実行し、より柔軟な単語ベクトルを獲得する
- 単語の順序情報は周期性のあるsin,cos関数でベクトル表現され、最初に各単語ベクトルに加算される
- 従来のseq2seqの逐次的処理と異なり、一度に複数の単語を並列化処理でき、計算効率が高い

イメージ図



BERT (Bidirectional Encoder Representations from Transformers)

BERTは大量のテキストデータから事前学習されたTransformerのエンコーダモデルであり、文章全体の文脈理解に優れており、文章分類や感情分析のタスクに適しています

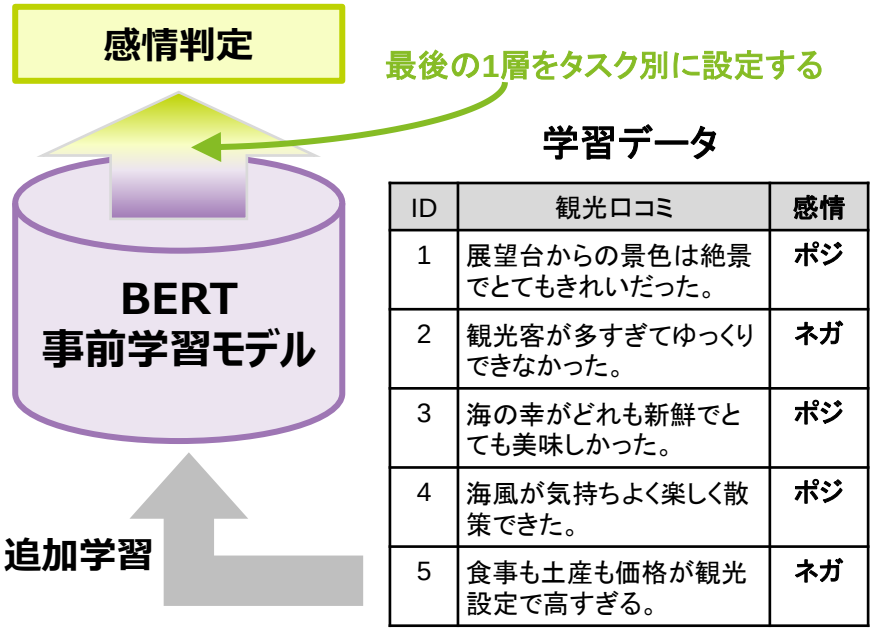
概要

- BERTはTransformerを使って大量のテキストデータを事前学習した汎用的な大規模言語モデルであり、2018年10月にGoogleによって発表され、2019年10月には検索エンジンに採用している
- Transformerのエンコーダ部分を使用したモデルで、文脈を捉えた単語のベクトル表現あるいは文章のベクトル表現を得ることができる
  - 文章全体の文脈理解において優れており、文章分類や感情分析などのタスクに適していると言われる
- Bidirectional（双方向）の意味は、Self-Attentionを実行するときに、各単語は同一文章中の左右にある他の全ての単語との関係性を捉えるということ
- 事前学習は自己教師あり学習で、一部の単語をマスクしその単語を双方向から予測するMLM（Masked Language Model）と、2つの文が連続するか否かを予測するNSP（Next Sentence Prediction）を実行し、文章全体の文脈理解能力を獲得する
- 事前学習モデルに追加の教師データでファインチューニングすることで、特定のタスクに応じてモデルが微調整され、少量の学習データでも高い精度が得られる

イメージ図

BERTのSelf-Attention

BERTのファインチューニング



学習データ

ID	観光口コミ	感情
1	展望台からの景色は絶景でとてもきれいだった。	ポジ
2	観光客が多すぎてゆっくりできなかった。	ネガ
3	海の幸がどれも新鮮でとても美味しかった。	ポジ
4	海風が気持ちよく楽しく散策できた。	ポジ
5	食事も土産も価格が観光設定で高すぎる。	ネガ

GPT (Generative Pre-trained Transformer)

GPTは大量のテキストデータから事前学習されたTransformerのデコーダモデルであり、テキストの生成に優れており、文章作成や要約作成、対話生成のタスクに適しています

概要

- GPTはTransformerを使って大量のテキストデータを事前学習した汎用的な大規模言語モデルであり、2018年6月にOpenAIによって発表された
- Transformerのデコーダ部分を使用したモデルで、入力の次にくる単語を予測することでテキストの生成ができ、文章作成、要約作成、対話生成、言語翻訳などのタスクに適していると言われる
- Self-Attentionを実行するときは、左側にある単語のみが使われ、一方向の関係性を捉える
- 事前学習は自己教師あり学習となり、テキストの次にくる単語を予測し、テキストの生成能力を獲得する
- 事前学習モデルを再利用する際は、特定のタスクの説明や指示を入力（プロンプト）として与えるが、shotと呼ばれるタスクを解く例（サンプル）を少数与えることで、ファインチューニングをすることなく様々なタスクに柔軟に対応できる（Few-Shot Learning）
- OpenAIは2019年にGPT-2を、2020年にGPT-3を発表し、2022年11月にはGPT-3.5をベースとした"ChatGPT"をリリースし、利用者は2ヶ月で1億人に達し、その性能の高さに世界は騒然とした

イメージ図

GPTのSelf-Attention

GPTのFew-Shot Learningのプロンプト例

Few-Shot 回答>> 感情：ポジティブ

```
口コミの感情を分析してください。
口コミ：朝市で食べたうに丼が美味しすぎた。　感情：ポジティブ
口コミ：ロープウェイの待ち時間が長すぎる。　感情：ネガティブ
口コミ：赤レンガ倉庫でお土産を買いました。　感情：ニュートラル
口コミ：元町は歴史的な建物が多く散策が楽しかった。
```

One-Shot 回答>> 英語：Where is the ropeway station?

```
日本語を英語に翻訳してください。
日本語：写真を撮っていただけませんか？
英語：Could you take our picture, please?
日本語：ロープウェイ乗り場はどこですか？
```

Zero-Shot

```
2泊3日で函館を一人旅するプランを考えて。
```

T5 (Text-to-Text Transfer Transformer)

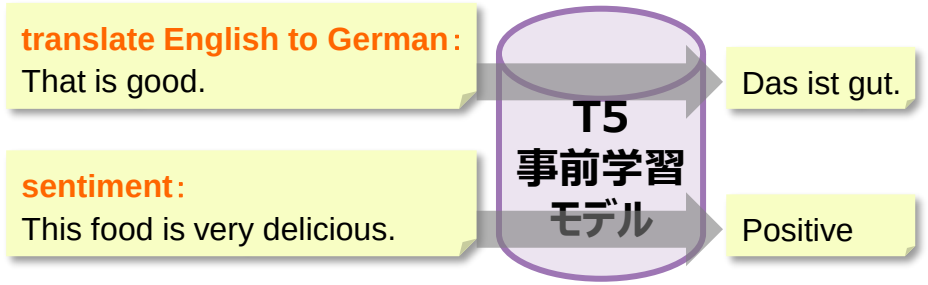
T5は大量のテキストデータから事前学習されたTransformerのエンコーダ-デコーダモデルであり、テキストをテキストに変換する処理で自然言語処理タスク全般に対応できます

概要

- T5はTransformerを使って大量のテキストデータを事前学習した汎用的な大規模言語モデルであり、2019年10月にGoogleによって発表された
- Transformerのエンコーダ-デコーダ構造を持つモデルで、従来のように個別タスクごとにモデルを構築するのではなく、あらゆる自然言語処理タスクをテキストからテキストに変換する同一フレームワークで扱える
- 事前学習は自己教師あり学習となり、エンコーダでテキストの一部をトークンで置換し、デコーダでテキスト全体の生成をしてトークンを復元することで、文脈理解能力とテキスト生成能力の両方を身につけさせる
- ファインチューニングでは、特定のタスク（翻訳、要約、分類、感情分析等）を指示する"プレフィクス"というラベルを付けたデータを学習することで、一つのモデルで様々なタスクに高い性能で対応できる
- GPTでもfew-shot learningで様々なタスクに対応可能だが、良質な例(shot)の提供が求められ、またGPTは入力の次にくる単語を予測する処理であるが、T5のように入力と出力の関係を強く捉えた生成ではない
- T5は比較的多くの計算資源が必要となる

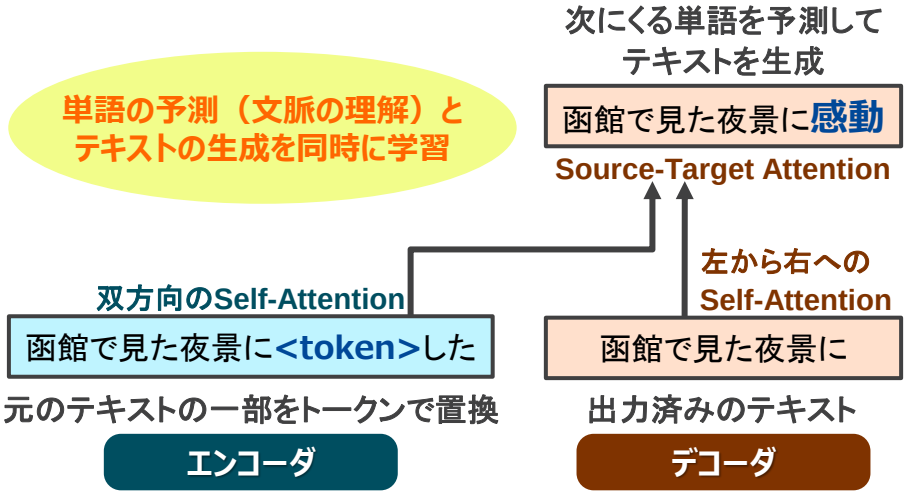
イメージ図

T5によるテキストからテキストへの変換



プレフィクス(タスクの指示)が与えられたテキストの入力から、そのタスクに対応したテキストを出力できる

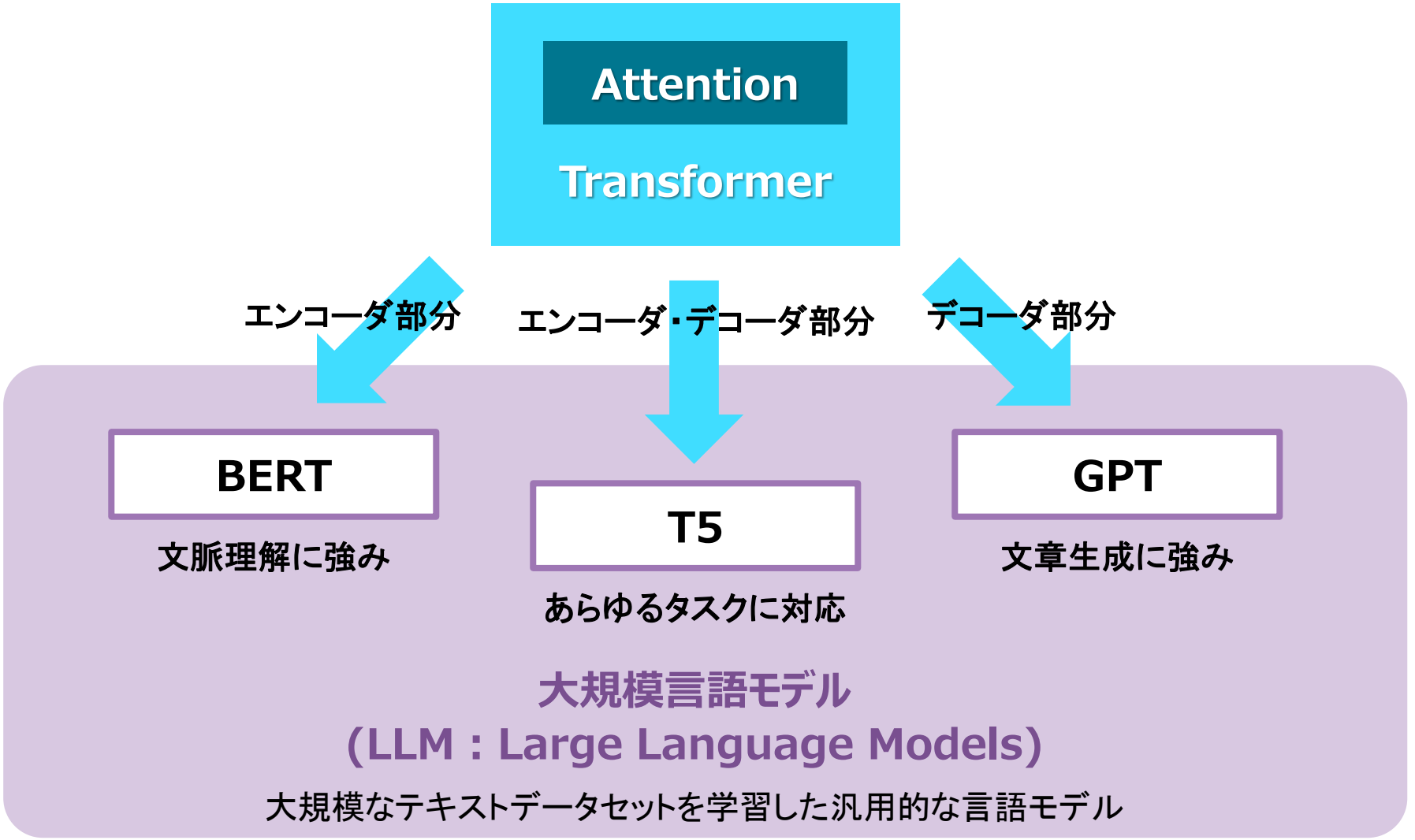
T5の事前学習におけるエンコーダとデコーダの処理



7. 大規模言語モデルとテキストマイニング

大規模言語モデルの位置づけ

BERTとGPTとT5は、Attentionの仕組みだけを用いたTransformerというアーキテクチャに基づき、大量のテキストデータから汎用的な言語特徴を学習した大規模言語モデルです



大規模言語モデルとテキストマイニングの対比

大規模言語モデルとテキストマイニングはどちらもテキストデータの分析と活用を目的とした手段という意味では共通しますが、それぞれの用途や特徴には大きな違いがあります

	大規模言語モデル	テキストマイニング
用途	**文脈の評価や文章の生成** （文章分類、感情分析、文章作成、要約作成、質問応答、対話生成、言語翻訳）	**特定のテキストデータの特徴の把握** （単語頻度の集計、単語のネットワーク化、単語のマップ化、属性別の特徴語把握）
適用範囲の性質	**汎用性** （言語の汎用的な特徴を事前学習し多様なタスクに適用可能）	**個別性** （特定のデータセットに存在する個別の特徴や傾向を把握可能）
課題解決の貢献	**直接的** （アウトプットそのものが課題解決において直接的な価値を提供）	**間接的** （アウトプットは課題解決のための意思決定に資する価値を提供）
アウトプットの形	**定性的** （テキスト形式による出力）	**定量的** （頻度の集計や統計解析による可視化）
情報処理の方式	**推論ベース** （事前学習した汎用モデルで確率を推論）	**事実ベース** （データ内の単語の出現頻度で分析）
表現の軸	**文脈** （単語間の相互関係性）	**単語** （個別単語の出現有無）
単語の認識単位	**トークン** （文字の出現パターンに基づいて結合された文字・サブワード・単語単位）	**形態素** （言語の文法ルールや辞書に基づいて分割された意味を持つ最小単位）
入力データの規模	**限定的なテキストデータ** （入力単語数に制限あり）	**大量のテキストデータ** （入力単語数に制限なし）

大規模言語モデルとテキストマイニングを相互補完的に活用することで、ビジネスの課題解決においてより有用なアプローチを形成できる可能性があります

大規模言語モデルをテキストマイニングの前処理に

大規模言語モデル → テキストマイニング

文章分類・感情分析

大規模言語モデルで文脈に基づく精度の高い文章分類や感情分析を行い、その結果にテキストマイニングを実行すれば、分類別の違いを単語ベースに理解できる

文章クラスタリング

大規模言語モデルで文章の特徴ベクトルをエンコードし、それを軸にクラスター分析してテキストマイニングを実行すれば、各クラスターの特徴を単語ベースに理解できる

多言語翻訳

多言語のテキストデータをテキストマイニングするとき、大規模言語モデルで同一言語に翻訳できる

辞書の候補語生成

テキストマイニングの辞書作りにおいて、大規模言語モデルを使って類義語の候補を生成できる

大規模言語モデルは一度に大量のテキストデータを処理することは得意でないため、対象のデータ量が多いときは、事前に分割するなどの工夫が必要である

テキストマイニングを大規模言語モデルの前処理に

テキストマイニング → 大規模言語モデル

深堀対象の要約

テキストマイニングの定量的な結果から、さらに深掘りしたい対象を絞り込み、その絞り込んだテキストデータに対して、大規模言語モデルで要約生成すれば、その対象の特徴を定性的に理解できる

深堀対象の細分化

テキストマイニングの結果から深堀対象としたテキストデータに対して、大規模言語モデルで文章分類や感情分析を行えば、深堀対象をさらに細分化して分類することができ、細かい傾向を理解できる

テキストマイニングを大規模言語モデルの前段階として活用する場合は、最初から大量のテキストデータを対象とすることができる

生成AIのみによるテキストデータ分析の危険

生成AIの回答は事前学習された汎用的な言語モデルで“推論”された情報で、不正確であったり意図しない生成となることもあるため、生成AIだけに依存した分析は危険です

生成AIでテキストデータを分析して感じた注意点

注意 ラベルの自動分類は不正確

- テキストデータにラベル（トピック）を付けて自動分類させたところ、最初はうまく分類できていたが、生成が進むにつれて、ラベルの統一性がなくなったり、プロンプトの指示に従わなかったり、分類精度も下がった
- 特に読み込むテキストデータの量が多いと精度が悪くなる印象がある
- Self-Attentionであっても長い文章で大量の単語があれば、遠い位置の単語のスコアは相対的に小さくなり、また近い単語ほど関連度が高いという一般的な傾向をモデルは学習しているはずである

注意 形態素解析はできない

- 生成AIで形態素解析を実行させたところ、正確な単語の抽出と頻度の集計はできなかった
- 正確さを求めるとPythonでMeCab（形態素解析エンジン）をインストールすることを求められた

注意 一度の誤りが影響し続ける

- 一度誤った生成されると、それを訂正したときは回答を正すが、対話を続けていると、前の誤りを踏襲していることがあった
- 過去の誤った生成結果にもSelf-Attentionがかけられるため、途中で訂正された内容があっても、それよりも前の誤情報の方でスコアが高くなってしまう場合があることも考えられる

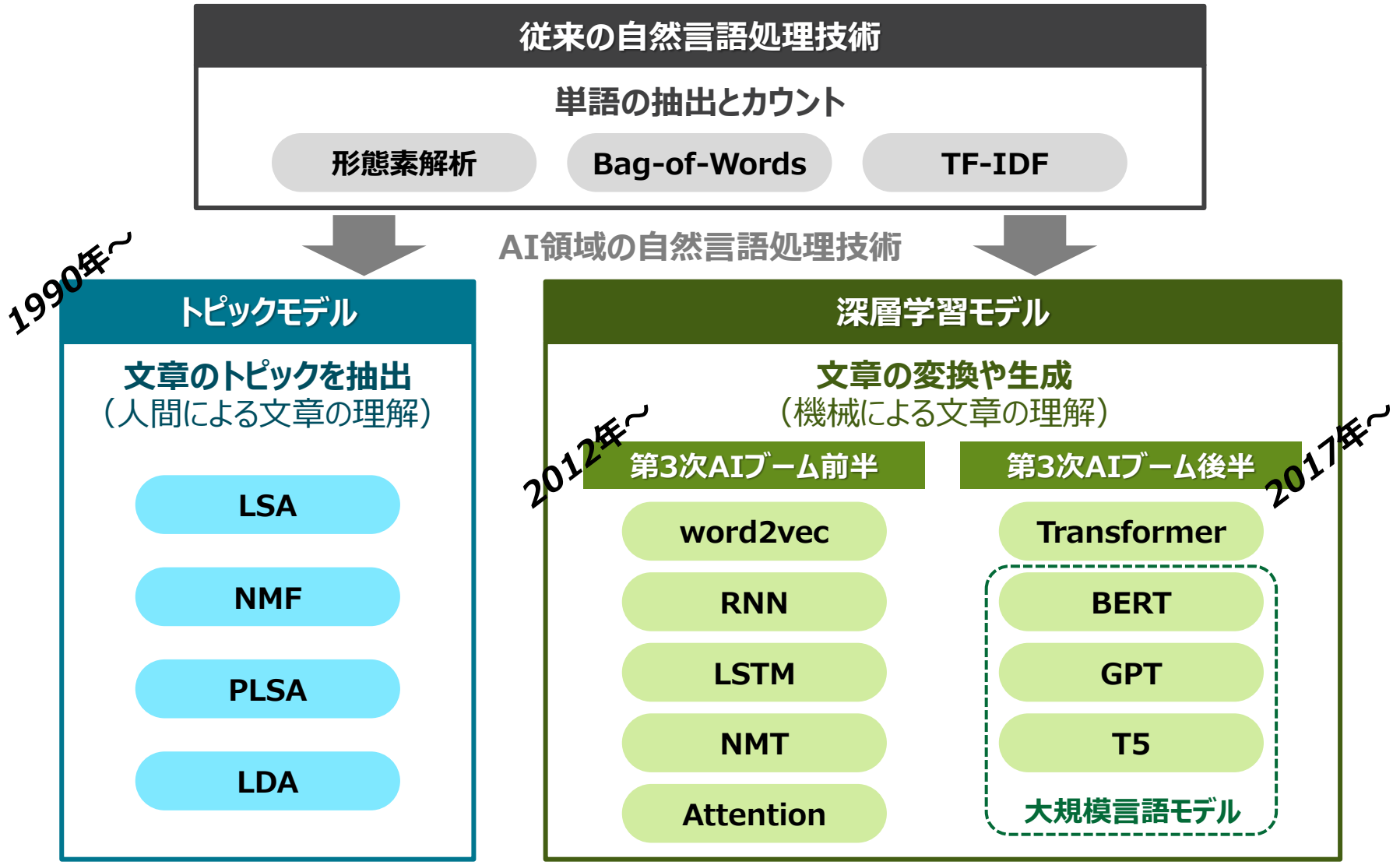
注意 要約生成で重要な点が省略される

- 要約を生成させれば、大量の文書でも分かりやすくまとめてくれるが、ビジネスインサイトになり得るような重要な気づきとなるような細かい内容が要約で省略されてしまうことがあった
- 大規模言語モデルは個別性よりも汎用性を重視されたモデルであるためと考えられるが、ビジネスインサイトとなる情報はしばしば個別性があるものである

8. まとめ

自然言語処理の技術整理

自然言語処理技術の各手法は、新旧はあっても明確な優劣があるわけではなく、固有の特徴があり、解決すべき課題にはどの手法が有効であるのか選択できることが重要です



書籍発刊のご紹介

10月31日に技術情報協会様から書籍「自然言語処理の導入と活用事例」が発刊され、私も一部の執筆を担当し、本日講演した内容も全て文章で解説しています

書籍情報

自然言語処理の導入と活用事例
ー情報検索、情報抽出、文書分類、テキスト要約ー

- 発刊予定：2024年10月31日
- 定価：88,000円（税込）
- 体裁：A4判　605頁
- ISBN：978-4-86798-049-1
- 出版社：株式会社技術情報協会
- URL：https://www.gijutu.co.jp/doc/b_2270.htm

執筆担当

- 第1章：近年の自然言語処理技術における各種手法の概要と特徴
- 第5章第10節：トピックモデルを応用したテキストデータの理解とインサイトの獲得

※私が執筆した原稿は書籍発刊後に弊社HPで公開予定
http://www.analyticsdlab.co.jp/seminar.html

資料に関するお問い合わせやコンサルティングの
ご相談は以下までお願いします。

analytics.office@analyticsdlab.co.jp

会社ホームページもご参考にしてください。
過去の講演・論文資料や技術解説も掲載しています。

http://www.analyticsdlab.co.jp/

株式会社アナリティクスデザインラボ

Analytics Design Lab