FERI INFORMATION

エグゼクティブのための生成 AI 講座

その２ ニューラルネットワークによる単語の分散表現

副島 豊（SBI 金融経済研究所・研究主幹）

LLM の発展をもたらした技術要素

イノベーションという概念を産み出したシュンペーターが実際に使った言葉は新結合であった。既存の技術やアイディアなどが結びつくことで新しいものが創造されるという考え方である。自然言語モデルのイノベーションにおいても、いくつかの重要な技術要素の発展とこれらの新結合があった。

その代表的なものとして、①言語の数値化・ベクトル化（分散表現あるいは埋め込み表現）への NN 学習手法の応用、②時系列データを扱える NN（RNN）を LLN に応用、③系列変換モデル（seq2seq）に対する Encoder/Decoder モデルの応用、④文意をとらえる Attention 機構の導入、⑤スケール則の発見（モデルやデータの規模拡大による性能向上の法則性）があげられる。まず、言語の数値化から説明する。

言葉を数値化し関係性を表現する

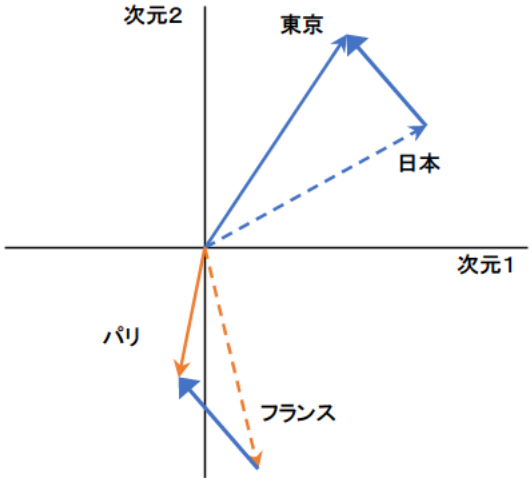
モデルは数字を使う。そのため自然言語モデルでは、まずは言葉を数値化する必要がある。単純なアプローチとして単語に ID を振る方法が考えられるが、膨大な量になるだけでなく、文章に内在する単語間の関係性をどのように効率的に表現するかという問題に直面する。仮に単語が 100 万語あって、単語 1 と残りの単語 2 ～単語 1,000,000 までとの関係を個々に数字で表現していくと、関係性を表現する情報は 100 万 x 100 万の行列となり非常に効率が悪い。当初は、実際の文章に出てくる単語の「離接」関係をそのまま情報として用い、単語をベクトル表現化するアプローチが採られた(1)。

エポックメイキングになったのは word2vec というモデルの登場である。これは、前後の複数の単語から間にある単語を推測する、あるいはある単語から前後の複数の単語を推測するという推論問題を NN に学習させ、推定されたパラメータから各単語をベクトル表現する手法である(2)。ベクトル化することで以下のような言葉の計算が可能となる。

東京－日本　＝　パリ－フランス

これは、概念上の情報処理ではなく、数字の計算として成立している。各単語が 2 次元空間に埋め込まれてベクトル化されているとしよう。日本やパリという単語は図 1 に示したベクトルとして表現できる。定式の左辺は図中の太い青矢印ベクトルであり、これは右辺（オレンジで示した 2 つのベクトルの差）と同一となっている。青矢印は首都という単語のベクトル表現に相当する。学習対象のコーパスに「日本の首都は東京である」「フランスの首都はパリである」といった文章が存在することで、5 つの単語の関係性を分散表現で表すことができる。

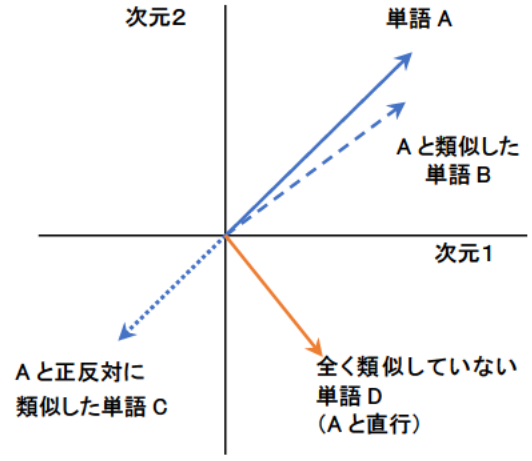
図１　パリ ＝ フランス＋（東京－日本）



出所：筆者作成

ここでは簡単化のために 2 次元とし、東京やパリという単語を図 1 中で（x_1, x_2）とベクトル表現したが、数十万の単語の関係を 2 次元に押し込むのは精度の面で困難であり、実際には数百～数千といった次元への圧縮が利用される。また、上の式も近似的に成立するに過ぎない。

図２　二次元空間で測った単語の類似性



出所：筆者作成

ベクトル化により単語の類似性の計測も図 2 のように可能となる。単語が数値化されているのでコサイン類似度や空間内での近傍関係を測る指標を活用することもできる。EC や映像音楽の配信などで各種の推薦サービスが提供されているが、これらは画像や商品特性をデジタル情報化（ベクトル化）したうえで、類似度を計算するという技術によって支えられている。

word2vec と転移学習

word2vec は転移学習を可能とした。単語の分散表現における転移学習とは、ある学習セットで作成したベクトル表現は、その学習セットが高い汎用性を有するものなら他の文章にも適用できる（転移可能である）ことを指す。日本語 Wikipedia のような大規模コーパスで学習した word2vec の結果は、再計算することなくそのまま一般的な文章に適用できる[(3)]。

見方を換えると、特殊な文章から成るコーパスには適用しにくく、学習し直しが必要となる。その場合も、言語の本質的な特性が変わっているわけではないので、ゼロから学習する必要はない。転移学習は、LLM においてもプレトレーニング（事前学習）とファインチューニングという実用上、非常に便利な特性をもたらしており、別稿で後述する。

なお、汎用性がある word2vec が多数公開されており、NN の学習なしでそのまま使うことができる（単語を入れるとベクトル表現に変換してくれる）。次元数は様々な設定が可能であるが、単語間の関係性を表現するためには数百次元程度のものが使われることが多い。

先に紹介した不効率な行列表現は 100 万 x 100 万であったが、単語数を同じ 100 万とす

るとベクトル表現（行列表現）は 100 万 x 数百まで圧縮される。単語の分散表現ではこうした次元圧縮によって、LLM の性能向上や計算コスト削減をもたらしている。

word2vec の限界

言語のベクトル表現に効率性をもたらした word2vec であったが、学習対象のコーパスに含まれる単語数が 100 万のように大きなものになると計算量は膨大なものとなる。このため、Negative sampling のような計算の効率化手法が必要とされた。

また、単語の隣接関係に基づく推論問題から算出されるため、位置関係より更に複雑な文脈の流れや、複数の意味や読み方を持つ文脈依存の多義語を扱うのが不得意であった。なによりも、word2vec は言語を生成するモデルではなく、単語をベクトル表現する装置であり、生成 AI の実用化はその後の LLM の発展があってのことであった。

LLM の発展には、単語が意味ある順序で並んでいるという文章の時系列データ的特性を表現するモデルが必要であった。株価の時系列データを並び替えると意味がなくなるのと同様、単語も文書内でランダムに並び替えることはできない。その意味で文書は単語の時系列データである。次の記事では、言語生成モデルとしての LLM 発展の起点のひとつとなった RNN を説明する。自然言語モデルに時系列データを扱える NN を適用することで言語生成モデルを作り出している。

なお、ベクトル表現のモデルは言語生成の LLM と並行して発展を続けており、有名なものとしては FastText や ELMo がある。ELMo では RNN の拡張版である LSTM（後述）を用いて、文章全体の情報を基づき単語をベクトル表現化し、かつ 1 つの単語に 3 つのベクトル表現（各々1024 次元）をもたせることで多義語対応などパフォーマンスを改善している。ELMo は他のベクトル表現モデルを補完する用途で提唱されたが、単独で利用することもできる。

ChatGPT で知られる OpenAI もベクトル表現に特化した LLM を開発し続けている。例えば、GPT-4 や 3 では、text-embedding-ada-002 というベクトル表現に特化した軽量高速のモデルが用いられている。LLM において言語や文書のベクトル表現化は、LLM の入り口（と出口における数値の言語復号）で行われる言語の数値化という極めて重要な役回りを果たしている。

```
•[1]: import os
      import openai
      from dotenv import load_dotenv
      load_dotenv()
      openai.api_key = os.environ["OPENAI_API_KEY"]

•[2]: def get_embedding(text_input):
          # テキストのベクトル変換はopenai.Embeddingで行う
          response  = openai.Embedding.create(
                          input = text_input.replace("\n",
                          model = "text-embedding-ada-002",
                      )
          # 出力結果の一部取得
          embeddings = response['data'][0]['embedding']
          return embeddings
```

出所：筆者作成

（その 3 に続く）

文末脚注

(1) 単語がどのような隣接関係をもって頻繁に登場するかを数えた共起行列や、その改善版の PMI（相互情報量：Pointwise Mutual Information）は、カウントベースのベクトル表現と呼ばれる。文章を構成する単語数に相当する高次元行列となるため、特異値分解という次元圧縮方法が使われる。

(2) NN へのインプットには各単語に割り当てられた one hot ベクトルが用いられる（単語 1 は{1,0,0,…}、単語 2 は{0,1,0,…}、単語 3 は{0,0,1,…}、以下同様）。これは、単語に ID を振るのと実質的に同じである。各ベクトルは直行関係にあるため、one hot ベクトルの組み合わせで他の one hot ベクトルを表現することはできない。このため NN のパラメータ（行列）に単語間の関係情報を集約することができる。推論問題はこの行列情報を得るための手段として設定されている。word2vec はシンプルな NN を用いており、推論結果表示のための確率表現以外のところでは行列演算を行っているにすぎない。図 1 に示したような単語ベクトルの線形演算が可能であるのは、このようなモデルの線形性に起因する。

(3) 一般には、転移学習とは、あるタスクのために学習させたモデルが、他のタスクにも転用可能であることを指す。例えば、文章の分類問題や要約作成、翻訳、賛成反対の判断、感情分析、対話生成など、LLM には様々なタスクが考えられるが、翻訳用に学習させた LLM が対話生成や要約作成にも高い能力を発揮するという汎用性を有する場合がある。あるいは、比較的少ない追加学習データや学習コストによって高い能力を持たせることが可能となる。この特徴は LLM の実用化に重要な役割を果たしている。