

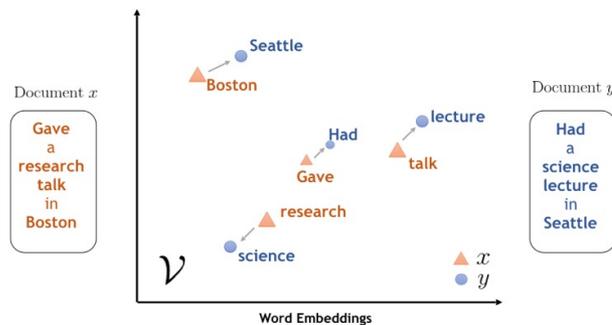
# 第3回目の補足資料

埋め込みとは？

# 埋め込みとは？

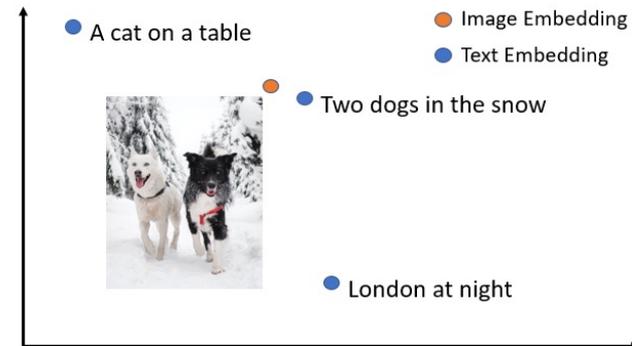
- 単語や画像をベクトル空間に位置付けること
  - Seattle→[0.1, 0.4, 0.3]、鶏の画像→[0.6, 0.3, 0.1]（実際は512次元とか）
  - 似た意味を持つ単語や画像が似たベクトルを持つようにする
- 最近では、単語・文章、画像、単語と画像、ネットワークのノードなど、いろいろなものが埋め込みによるベクトルとして表現し分析に用いられることが多い
- 埋め込みによりベクトルとして表現できると何が嬉しいか？
  - 大量のデータの中から類似検索やレコメンドができるようになる
  - 機械学習モデルの特徴量として使える
- 大規模な事前学習モデルを有効に活用することで、自らは学習せずにベクトル化できる

## 単語の埋め込み



<https://www.ibm.com/blogs/research/2018/11/word-movers-embedding/>

## 単語と画像の埋め込み



<https://medium.com/nlplanet/two-minutes-nlp-sentence-transformers-cheat-sheet-2e9865083e7a>

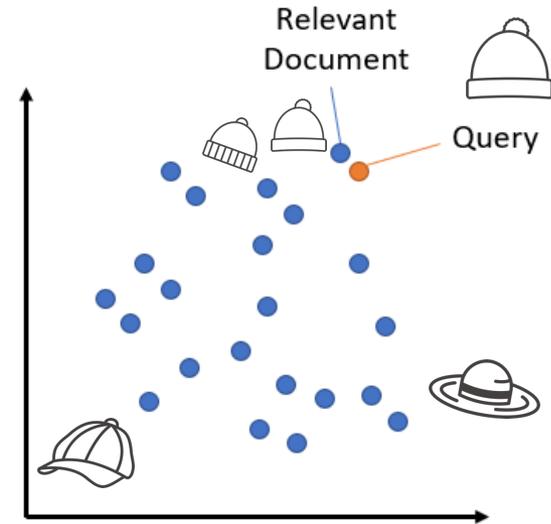
# ZOZOの類似画像検索（画像特徴量の抽出）

「類似アイテム検索機能」の使い方



<https://www.itmedia.co.jp/news/articles/1908/26/news068.html>

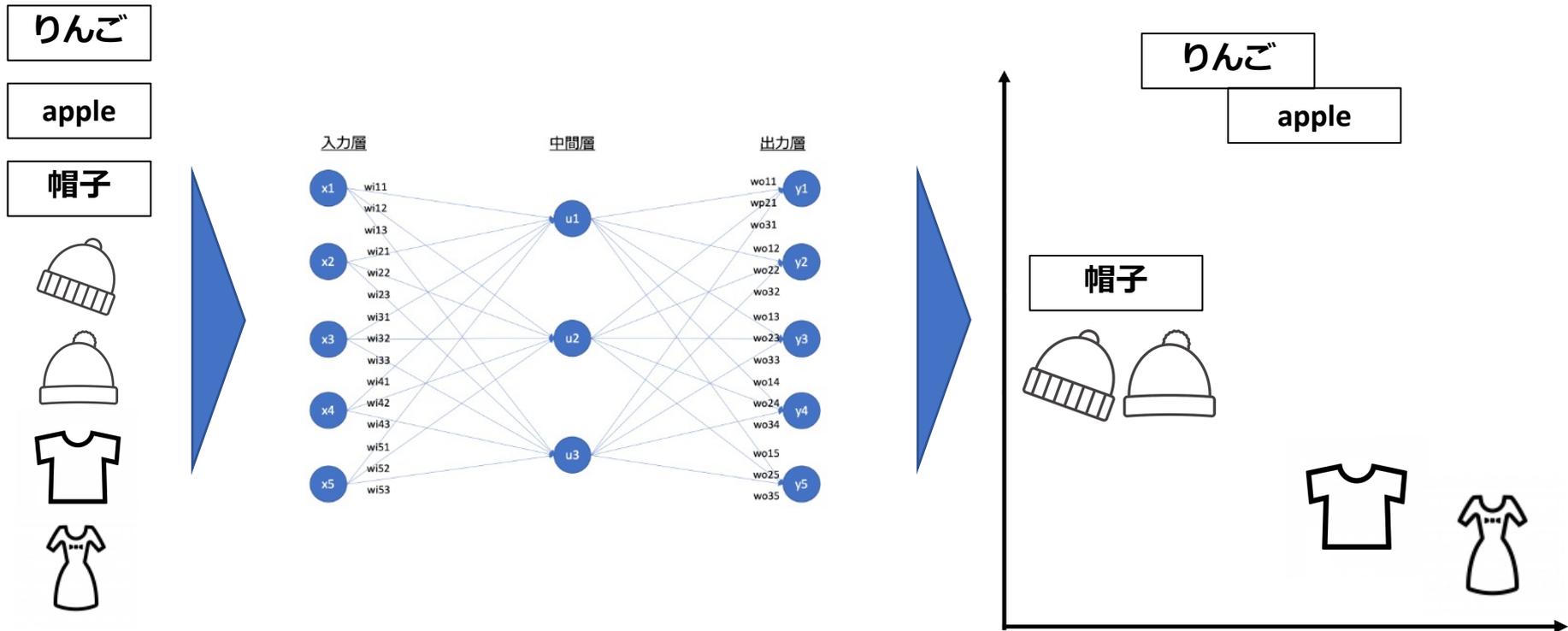
## 画像の検索



# 埋め込みと学習

- 文字列や画像である $x$ が、関数 $f()$ （今は多くの場合、深層学習モデル）を通して、ベクトルに変換される
- 近づきたい $x$ （と遠ざけたい $x$ ）を教師データとして作りそれを深層学習モデルに学習させる

$$X \xrightarrow{v = f(x)} V$$



# word2vecによる埋め込みベクトル の作成過程の解説

# word2vecをイメージで理解する

- window size = 周辺何個までの単語を考えるか？
- 黄色の単語が、緑の単語の平均のベクトルとなるイメージ

window size=0) I want to eat an apple everyday.  
window size=1) I want to eat an apple everyday.  
window size=2) I want to eat an apple everyday.  
window size=3) I want to eat an apple everyday.

- 周辺単語が類似する単語 → 類似したベクトル、として学習される

I want to eat an apple every morning.  
I like to eat an apple.

• • •

I want to eat an orange every morning.  
I like to eat an orange.

• • •

# BERTによる文脈を考慮した埋め込みベクトルの作成

# BERTとは

事前モデル作成のとき

- Masked ML
- Next Sentence Prediction

Fine-tuningのとき

- 分類タスクは全結合層などを結合

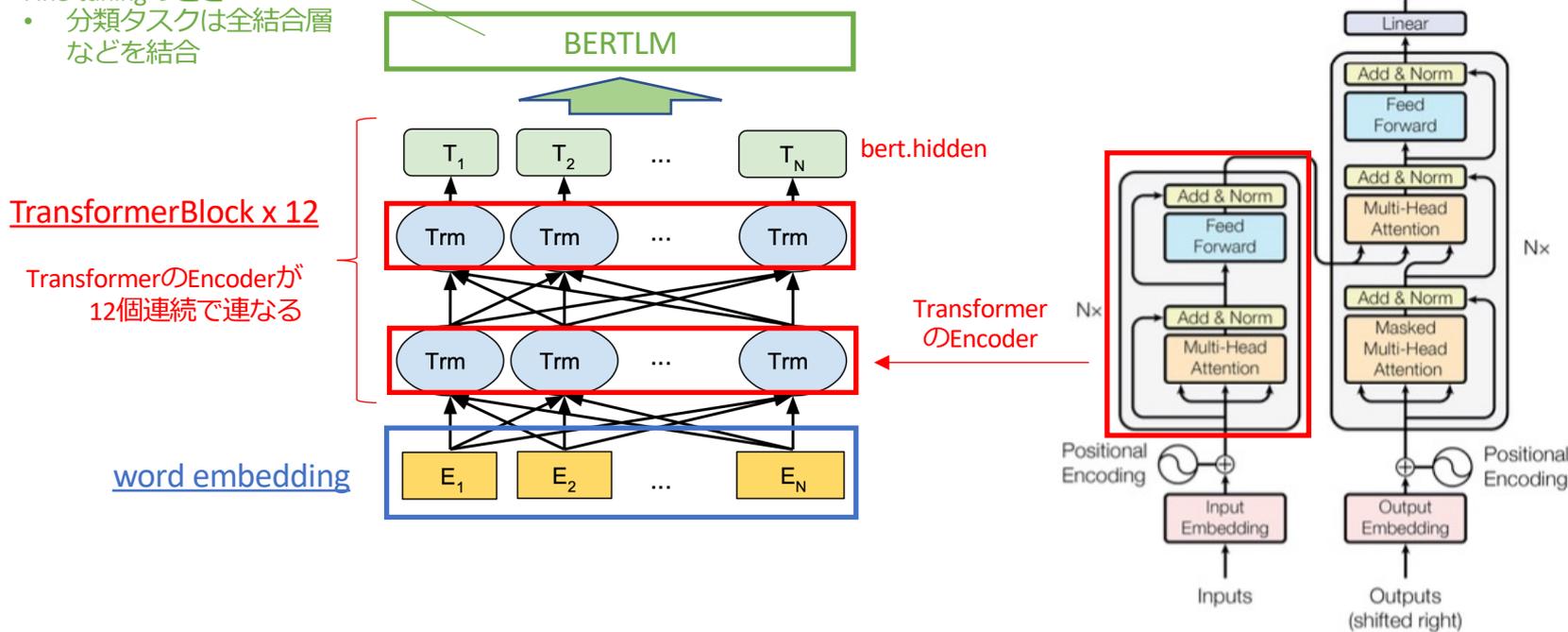


Figure 1: The Transformer - model architecture.

# word Embeddingについて

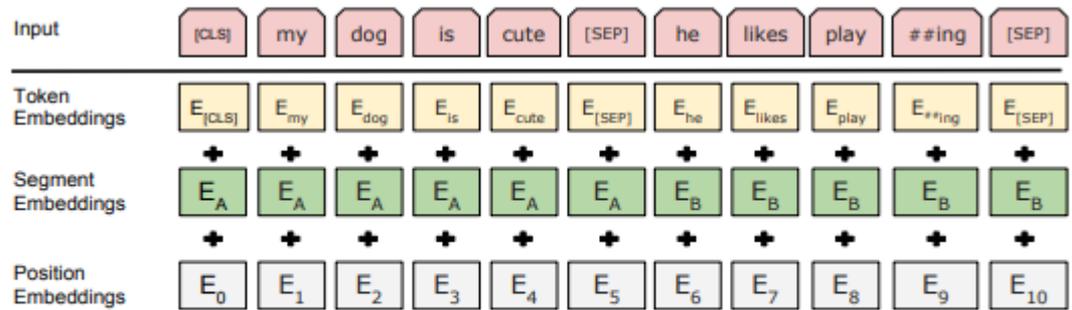
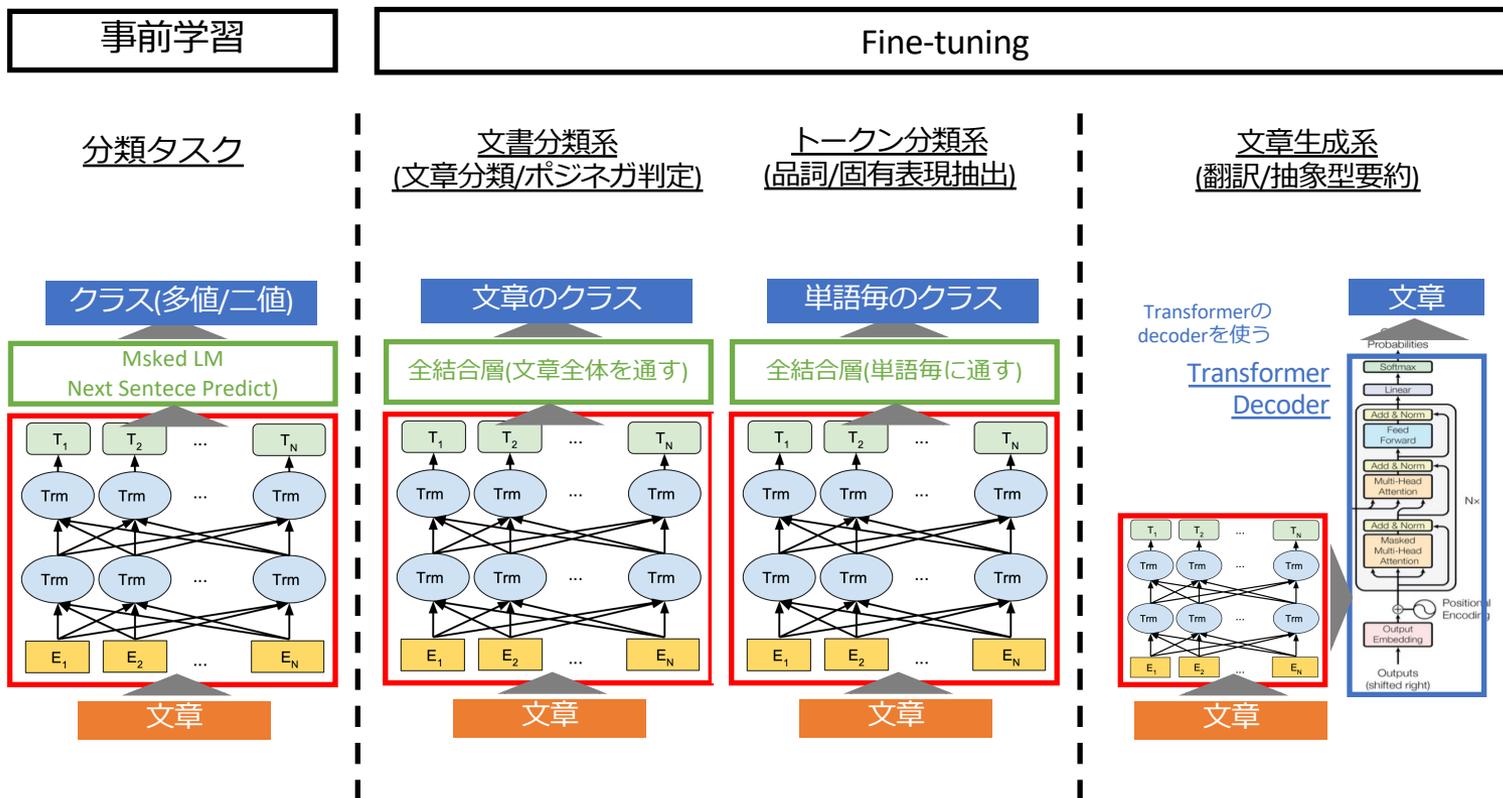


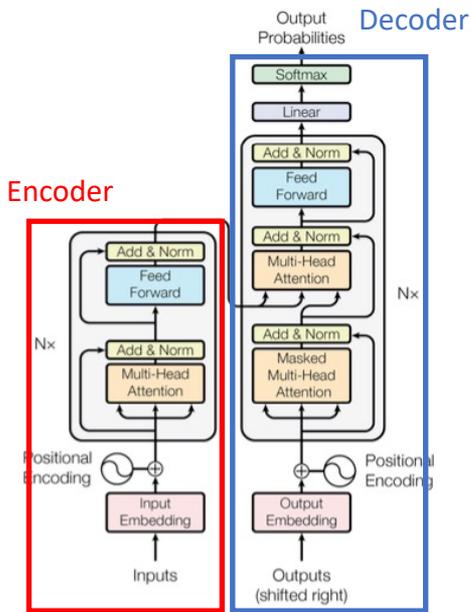
Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

# BERTの各種タスクへの適用

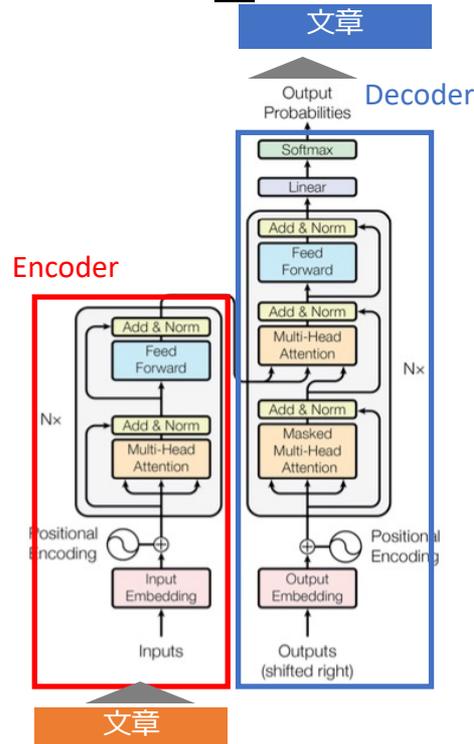


# Transformerの各種タスクへの適用

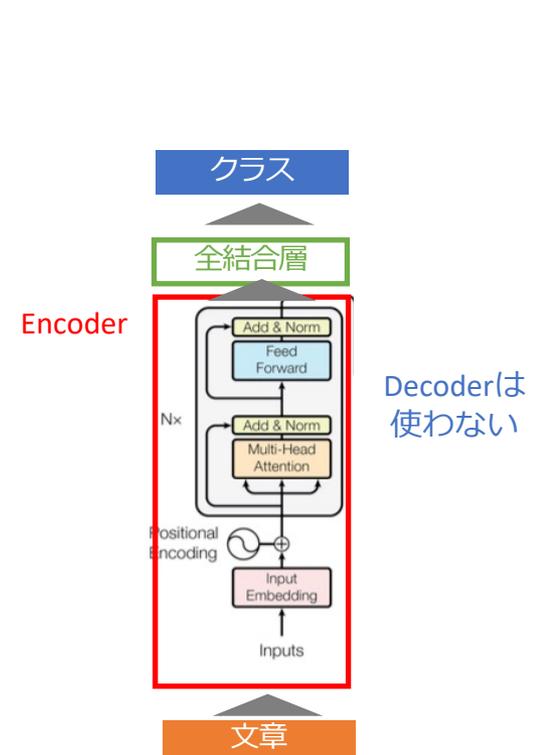
## Transformer



## 文章生成系 (翻訳/抽象型要約)



## 分類系 (文章分類/ポジネガ判定)

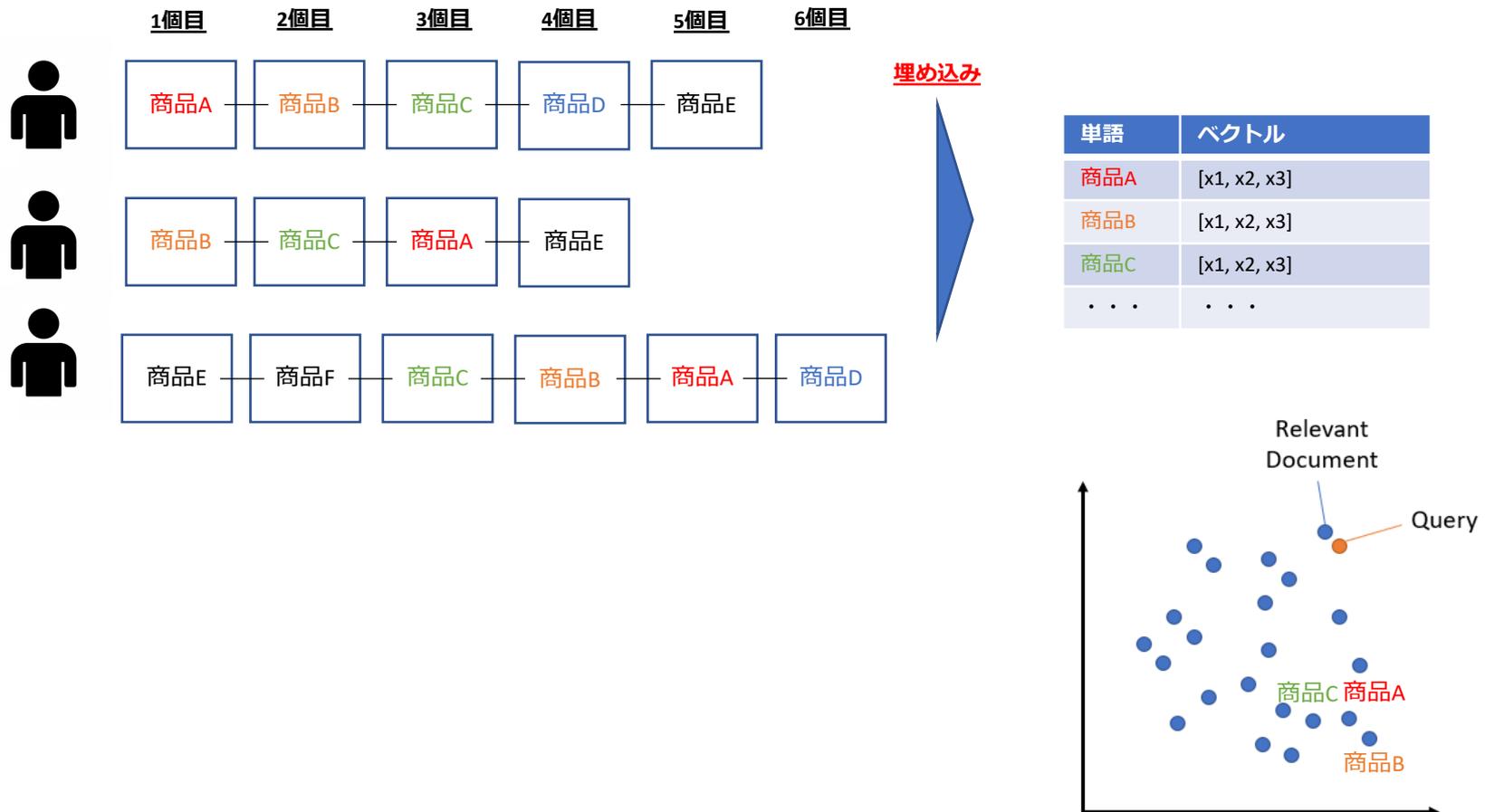


# アナロジーを使ったword2vecの埋め込みの活用

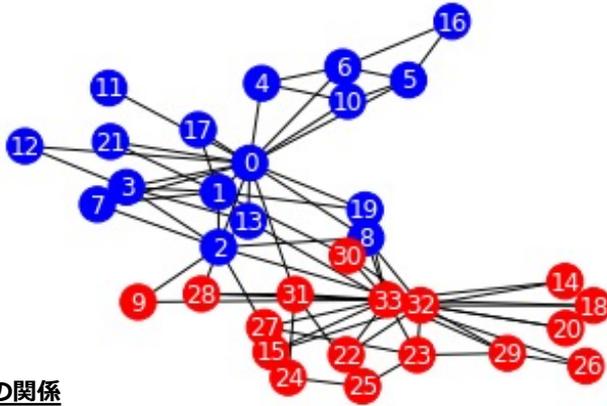
- 商品の埋め込み (item2vec)
- グラフの埋め込み (DeepWalk)

# アイテムの埋め込み (item2vec)

- ECサイトにおけるアイテムの購入履歴をword2vecをすることで、アイテムをベクトルとして埋め込む
- 購買のされ方が似ている商品はベクトル空間上で近い座標に位置する



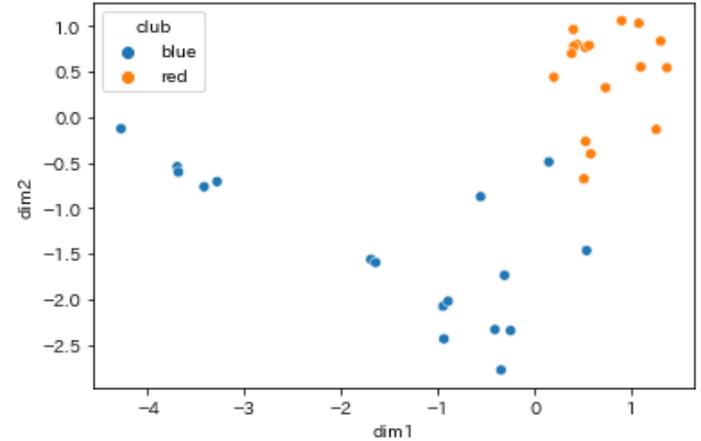
# ネットワークの埋め込み (DeepWalk)



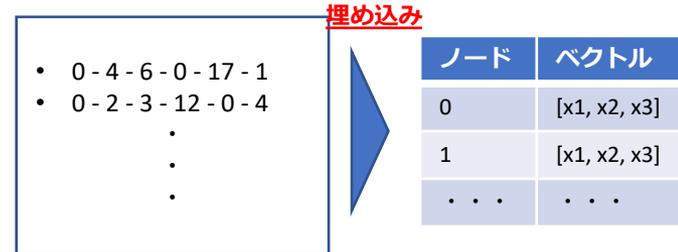
エッジの関係

[[0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (0, 8), (0, 10), (0, 11), (0, 12), (0, 13), (0, 17), (0, 19), (0, 21), (0, 31), (1, 2), (1, 3), (1, 7), (1, 13), (1, 17), (1, 19), (1, 21), (1, 30), (2, 3), (2, 7), (2, 8), (2, 9), (2, 13), (2, 27), (2, 28), (2, 32), (3, 7), (3, 12), (3, 13), (4, 6), (4, 10), (5, 6), (5, 10), (5, 16), (6, 16), (8, 30), (8, 32), (8, 33), (9, 33), (13, 33), (14, 32), (14, 33), (15, 32), (15, 33), (18, 32), (18, 33), (19, 33), (20, 32), (20, 33), (22, 32), (22, 33), (23, 25), (23, 27), (23, 29), (23, 32), (23, 33), (24, 25), (24, 27), (24, 31), (25, 31), (26, 29), (26, 33), (27, 33), (28, 31), (28, 33), (29, 32), (29, 33), (30, 32), (30, 33), (31, 32), (31, 33), (32, 33)]

埋め込み



- 「単語の集まりが文章」 ⇔ 「ノードの集まりがグラフ」
- ネットワークの関係が似ているノードはベクトル空間上で近い座標に位置する

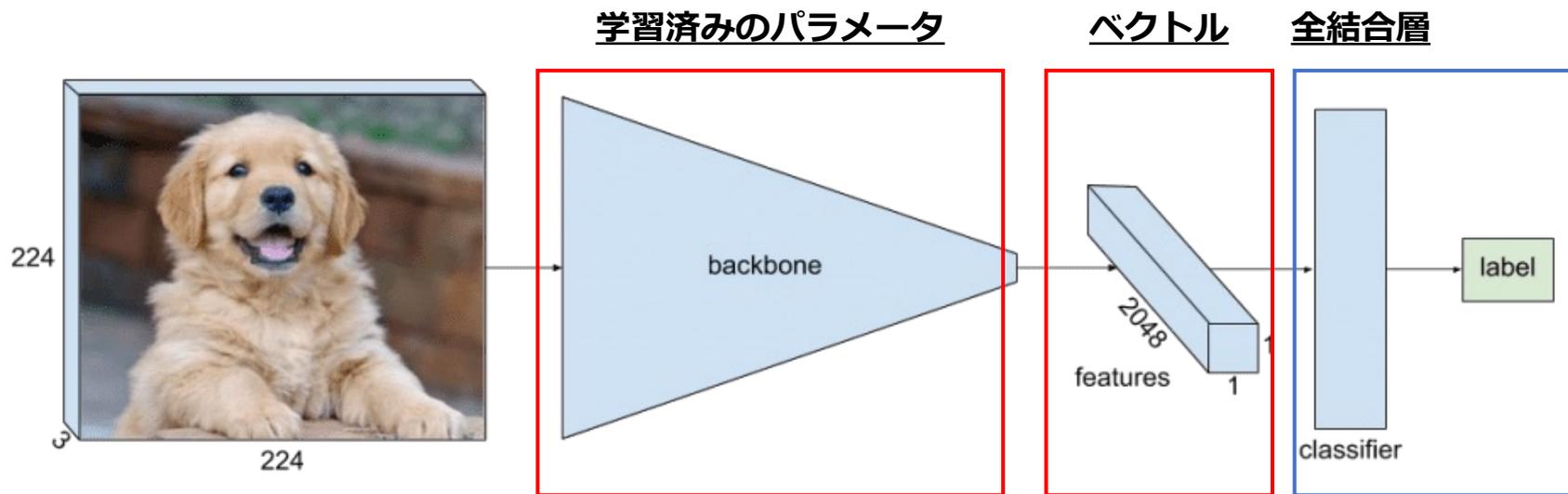


- 文章は一文がたくさんある「Gave a research talk in Boston、 Had a science lecture in Seattle」
- グラフでは、隣り合うノードをランダムで選択して短いノードのつながりをたくさん作る (random walk)

画像の埋め込み (image  
embedding)

# 画像特徴量の取得

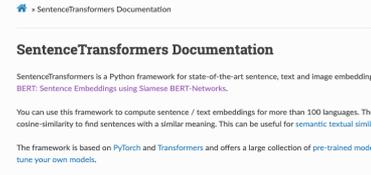
- CNNの最後の全結合層（classifierレイヤー）の一つ手前の値（ベクトル）を使う
- 普通は全結合層の出力が、学習した各ラベルの出現確率になっている



<https://learnopencv.com/t-sne-for-feature-visualization/>

# Sentence Transformerと事前学習済みモデルの利用

# Sentence Transformer (Sentence BERT)



- BERTを2つの文章に適用して類似・非類似を学習させたもの
- 2つの文をSentenceTransformerネットワークに渡し、文の埋め込み $u$ と $v$ を取得します。次に、 $u$ 、 $v$ 、および $|u-v|$ を連結します。1つの長いベクトルを形成します。このベクトルはソフトマックス分類器に渡され、3つのクラス(含意、中立、矛盾)が予測されます。
  - <https://www.sbert.net/examples/training/nli/README.html>
  - <https://arxiv.org/pdf/1908.10084.pdf>

文A (前提)	文B (仮説)	ラベル
複数の男性がプレーするサッカーゲーム。	スポーツをしている男性もいます。	含意
笑顔の年配の男性と年下の男性。	床で遊ぶ猫たちを見て、2人の男性がにっこり笑っている。	中性
東アジアのある国で人物の制服を検査する男性。	男は寝ている。	矛盾

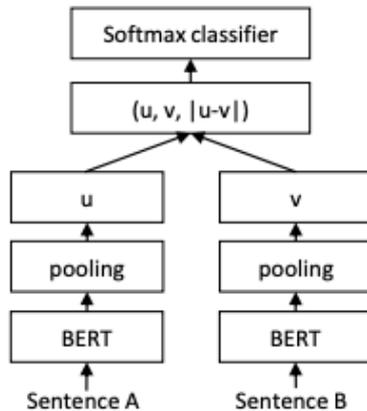


Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

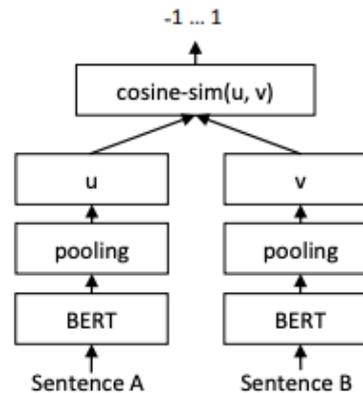


Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

# Sentence Transformerを使った単語 と画像の埋め込み

# CLIP (Contrastive Language-Image Pre-Training)

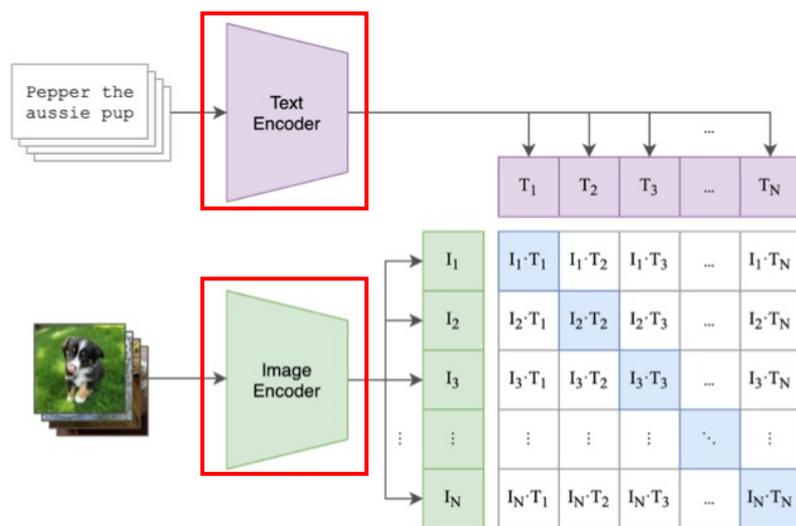
## • 参考

- <https://github.com/openai/CLIP>
- <http://cedro3.com/ai/clip-search/>

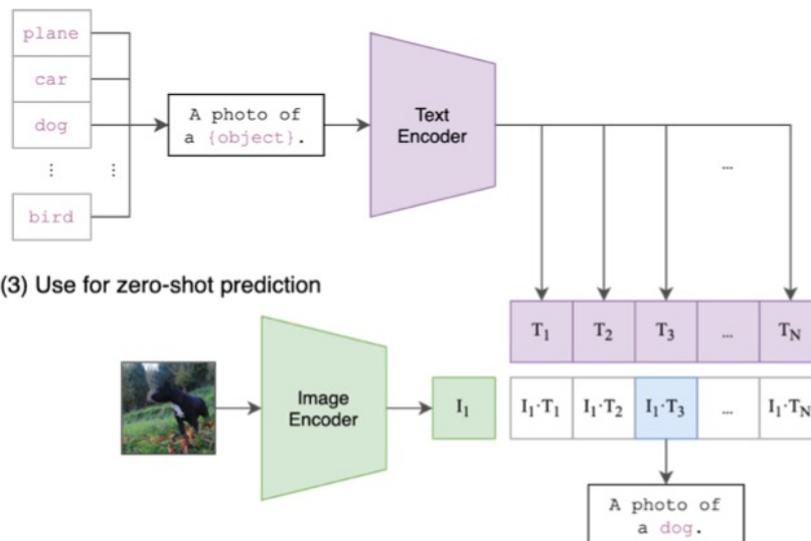
## • テキストと画像を同時に事前学習する方法

- テキストをエンコードしてベクトルしたものと、画像をエンコードしてベクトルにしたものの内積を最大／最小するようにエンコーダーを学習する
- 学習した「テキストエンコーダ」「画像エンコーダ」を使うことで、類似の意味を持つテキストと画像を近いベクトルで表現できる

(1) Contrastive pre-training



(2) Create dataset classifier from label text



(3) Use for zero-shot prediction