4-5 自然言語処理

東京大学　数理・情報教育研究センター

2020年4月28日

2024年4月28日改訂

概要

・文書（テキスト）をデータとして処理するため，テキストを分割する分かち書きと形態素解析について学びます．またベクトルを用いてテキストを数的に処理するための方法について学びます．

本教材の目次

コーパス

- テキスト解析では，解析の対象をよく表すようなテキストデータのサンプルを収集します．
  - テキストデータには，記事，論文，特許，カルテ，ウェブページなどがあります．

- 収集されたテキストデータのサンプル集合をコーパスと呼びます．コーパスは解析の目的に応じて収集・構築します．
  - 例：国立国語研究所の日本語コーパス
    - 話し言葉・会話コーパス
      - **https://pj.ninjal.ac.jp/corpus_center/**
    - ウェブコーパス
      - **https://pj.ninjal.ac.jp/corpus_center/nwjc/**

- ウェブAPIを利用してテキストデータを収集してコーパスを構築することもできます．
  - 例：国会議事録検索のAPI
    - **https://kokkai.ndl.go.jp/api.html**

分かち書き（単語分割）

- テキストを「トークン」と呼ばれる表現要素の最小単位の集合に分割することを分かち書き（単語分割）と呼びます．単語に限らずテキストを構成する記号や数字などもトークンとなりえます．

- 英語の場合，テキストをスペースやカンマで区切れば単純な分かち書きをすることができます．

  ‘Beware the ides of March’
  →
  [‘ Beware ’, ‘ the ’, ‘ ides ’, ‘ of ’, ‘ March ’]

分かち書き（単語分割）

- 日本語の場合は句読点以外はスペースやカンマのような明確な区切りはテキストにないため，何らかの処理によりテキストを分割する必要があります.

- 国会図書館が作成する書誌データの標目の読みに用いるための分かち書きの基準として以下のように示されています.

  “分かち書きは，検索語となる自立語を対象とする．日本語の場合は，日本語として不自然でない意味のまとまりで分かち書きを行う．外来語の場合は，当該言語の単語分割により分かち書きを行う．”

  https://www.ndl.go.jp/jp/data/catstandards/yomi/word_division_Apr2008.html

形態素

- 言語学では意味を持つ表現要素の最小単位は形態素と呼ばれます．より直感的には形態素は，名詞，動詞，形容詞，副詞，前置詞の品詞や語形・活用形などの文法的役割を表す語のクラスを表します．

- テキストをトークンに分割し，各トークンにこのような品詞や語形・活用形などの情報を付与する処理を形態素解析と呼びます．

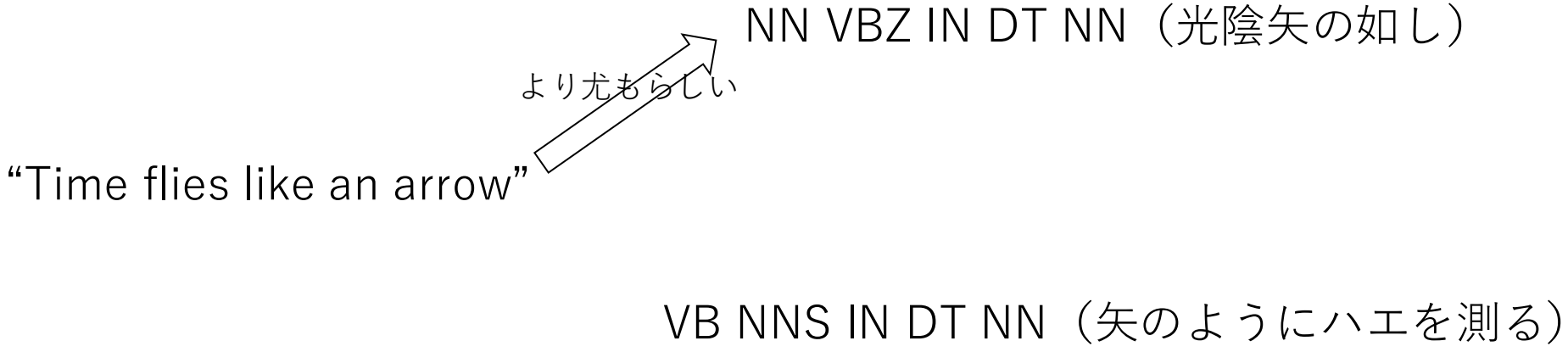
- 形態素を表すタグを品詞（POS: Part of Speech）タグと呼びます．

  - 例えば，名詞であれば単数名詞はNN，複数名詞はNNS，固有名詞はNPというタグでそれぞれ表されます．

  - その他，VB:動詞現在形，JJ:形容詞，RB:副詞，など
    - 新聞記事コーパスのPenn Treebankに付加されたPOSタグの例
      - **https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html**

形態素解析

- 形態素解析では，与えられたトークンの系列に対して尤もらしい形態素タグの系列を予測するという系列ラベリング問題を解きます.



形態素解析

形態素解析の例：

入力：「テキストもデータであることを理解する．」

形態素解析の結果：

テキスト	名詞,一般,*,*,*,*,テキスト,テキスト,テキスト
も	助詞,係助詞,*,*,*,*,も,モ,モ
データ	名詞,一般,*,*,*,*,データ,データ,データ
で	助動詞,*,*,*,特殊・ダ,連用形,だ,デ,デ
ある	助動詞,*,*,*,五段・ラ行アル,基本形,ある,アル,アル
こと	名詞,非自立,一般,*,*,*,こと,コト,コト
を	助詞,格助詞,一般,*,*,*,を,ヲ,ヲ
理解	名詞,サ変接続,*,*,*,*,理解,リカイ,リカイ
する	動詞,自立,*,*,サ変・スル,基本形,する,スル,スル
.	記号,句点,*,*,*,*,．,．,．

例えば，形態素解析の結果から
「テキスト」，「データ」，「こと」，「理解」
を抽出して，テキストの名詞を対象に解析することができます．

形態素解析

- 日本語の形態素解析では，辞書を参照しながらテキストから単語を取り出しそれらの連接の可能性を確認していきます．このためには，以下の辞書を用います．
  - 単語辞書：単語の品詞，読みや活用形を定義
  - 連接可能性辞書：連接可能な2つの単語または品詞・活用のタイプを定義

- 形態素解析の単語辞書に登録されていない単語を，未定義（または未知）語と呼びます．未定義語は多くの場合，固有名詞（人名，地名，組織名など），専門用語，新語や造語です．

形態素解析

- 単語辞書に新たに単語を追加する，または独自にユーザ定義辞書を作成する場合は，以下のように単語の情報を定義します.

見出し語	読み	品詞	活用型	活用形	基本形
人工知能	じんこうちのう	名詞	-	-	-
走る	はしる	動詞	五段・ラ行	終止形	走る

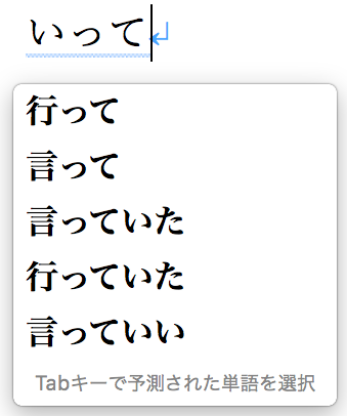
かな漢字変換

- かな漢字変換では，入力された「かな」のべた書きテキストを解析して，単語の区切りを見つけ出し，必要な部分を漢字に変換します．
- かな漢字変換において，単語の区切りを見つける際には，形態素解析の考え方を応用することができます．
- 一方，かなから漢字へ変換する際は，テキストの意味や文脈に応じて漢字を選択する必要があります．
  - 単純な漢字変換はあらかじめ決められた規則をもとに漢字候補の優先度を計算することで行います．
  - 単語の使用頻度，単語間の共起関係などの情報をもとに，よりテキストの意味や文脈を考慮した漢字変換を行うこともできます．

かな漢字変換

- かな漢字変換を行う日本語入力システムとしてIME（インプットメソッドエディタ）があります．最近では入力を予測しながら支援する入力予測機能を利用することもできます．

IMEによるかな漢字変換



入力予測機能

n-gram

- テキスト解析では，コーパスなどの大規模なテキスト集合を用いてその対象のテキストデータの性質をよく説明できるような統計的なモデルを構築することがあります.

- 単純な統計モデルにn-gram（nグラム）があります.

  - 文字n-gramでは，1文字，2文字（バイグラム），3文字（トライグラム），といった連続するn文字を単位としてコーパス内のそれらの頻度を数えます.
    - 「テキスト解析」のバイグラム
      - テキ，キス，スト，ト解，解析
    - 「テキスト解析」のトライグラム
      - テキス，キスト，スト解，ト解析

  - 頻度を数える単位は文字に限らず，単語や品詞を用いることもあります.

n-gram言語モデル

- n-gramは，文や表現の出現確率（文や表現が使われる確からしさ）を与える言語モデルに使われます.
- 言語モデルにより文字あるいは単語列に対して確率を与えることでその言語らしさ（対象言語における自然さ）を推定できます.
- 例えば単語を用いたn-gram言語モデルでは，n個の単語の連接をもとに文や表現の出現確率をモデル化します.
  - このとき，各単語は直前のn-1語に依存して出現するとして，コーパスにおける頻度情報をもとに出現確率を計算します.
- このような言語モデルは音声認識や機械翻訳などさまざまな応用においてより適切な出力を選択するための基準として利用されています.
- また，言語モデルは，直前のn-1語に依存して単語を確率的に選択することを繰り返すことで，単語の系列としてテキストを生成することにも応用することができます.

文書単語行列

- 形態素解析などの処理によりテキストを分割すると，以下のように複数のテキスト（文書）を表（テーブル）の形で表すことができます．この表を文書単語行列と呼び，行はテキスト，列は単語に対応します．

- 文書単語行列の値は，あるテキスト（行）にある単語（列）が現れる（1）・現れない（0）の2値，またはテキスト中の単語の頻度（出現頻度）で表します．

	言語	**AI**	**解析**	**データ**
言語を解析するAI	1	1	1	0
言語データを解析する	1	0	1	1
言語もデータである	1	0	0	1

*各テキストの名詞のみを抽出

文書単語行列

- 文書単語行列の列に対応する単語としては，以下に留意しながら，なるべくテキストで重要と考えられる単語を用います.
  - 助詞や助動詞などの機能語を用いるか
  - 高頻度で出現する一般的な単語を用いるか
  - 極端に出現頻度が低い単語を用いるか

- 文書単語行列の値は，出現頻度だけでなくテキストにおけるその単語の重要度を数値化したもので表すこともあります.
  - 代表的な重要度としてtf-idfがあります

文書単語行列

- tf-idf
  - tf-idfは単語の出現頻度（tf）にその単語を含むテキスト数の逆数（idf）を掛け合わせた値です.
  - idfは具体的には以下で定義されます.
    - idf=$\log_2$(すべてのテキスト数/単語を含むテキスト数)+1
      *値が大きくなりすぎないようにlog をとります

  - 例えば，「言語を解析するAI」の単語「AI」のtf-idf値は，tfが1，idfが$\log_2(3/1)+1$のため，$1\times(\log_2(3/1)+1)=2.585$
  - 「言語もデータである」の単語「データ」のtf-idf値は，tfが1，idfが$\log_2(3/2)+1$のため，$1\times(\log_2(3/2)+1)=1.585$

	言語	**AI**	**解析**	**データ**
言語を解析するAI	1	2.585	1.585	0
言語データを解析する	1	0	1.585	1.585
言語もデータである	1	0	0	1.585

文書（テキスト）ベクトル

- 文書単語行列の各行は，その行に対応するテキストをそのテキストにおける各単語の重要度を要素とするベクトル（文書ベクトル）で表したものとみることができます.
  - 例：テキストにおける各単語の出現頻度を重要度とすると，「言語」，「AI」，「解析」，「データ」の単語のベクトル空間において，各テキストのベクトルは以下のようになります.

	言語	**AI**	**解析**	**データ**
言語を解析するAI	1	1	1	0
言語データを解析する	1	0	1	1
言語もデータである	1	0	0	1

- テキスト1:「言語を解析するAI」 → $d_1 = (1, 1, 1, 0)$
- テキスト2:「言語データを解析する」 → $d_2 = (1, 0, 1, 1)$
- テキスト3:「言語もデータである」 → $d_3 = (1, 0, 0, 1)$

文書（テキスト）間類似度

- テキスト間の類似度は，各テキストを表すベクトル間の類似度として計算することができます．
  - ベクトル間の類似尺度して，コサイン類似度を用いることができます．
  - コサイン類似度は，2つのベクトルの内積をそれぞれの大きさ（ベクトルのノルム）で割ったものです．
  - コサイン類似度が高いほどそれらのテキストは関連度が高いと考えられます．

ベクトル

$$x = (x_1, x_2, \ldots, x_n) \qquad y = (y_1, y_2, \ldots, y_n)$$

内積

$$x \cdot y = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$$

ベクトルの大きさ（ノルム）

$$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

コサイン類似度 $\cos\theta = \frac{x \cdot y}{\|x\|_2 \|y\|_2}$ （θ は2つのベクトルのなす角）

文書（テキスト）間類似度

- 文書間類似度の例
  - テキスト1:「言語を解析するAI」 $\rightarrow d_1 = (1, 1, 1, 0)$
  - テキスト2:「言語データを解析する」 $\rightarrow d_2 = (1, 0, 1, 1)$
  - テキスト3:「言語もデータである」 $\rightarrow d_3 = (1, 0, 0, 1)$

  - テキスト1とテキスト2のコサイン類似度

$$\frac{\mathbf{d_1} \cdot \mathbf{d_2}}{\|\mathbf{d_1}\| \|\mathbf{d_2}\|} = (1 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 1) / (\sqrt{1^2 + 1^2 + 1^2} \times \sqrt{1^2 + 1^2 + 1^2}) = \frac{2}{3} = 0.666...$$

  - テキスト1とテキスト3のコサイン類似度

$$\frac{\mathbf{d_1} \cdot \mathbf{d_3}}{\|\mathbf{d_1}\| \|\mathbf{d_3}\|} = (1 \times 1 + 1 \times 0 + 1 \times 0 + 0 \times 1) / (\sqrt{1^2 + 1^2 + 1^2} \times \sqrt{1^2 + 1^2}) = \frac{1}{\sqrt{6}} = 0.408...$$

- この結果から，「言語」，「AI」，「解析」，「データ」の単語のベクトル空間においては，テキスト1はテキスト3よりもテキスト2と類似していると言えます.