# 自然言語処理
# 第2回

中田尚

# 自然言語とは（↔人工言語）

- 人間が日常的に使用している/いた言語
  - 文法よりも先に言語が存在している
  - 地域/時代によって意味や文法が変化することもある


- 例：以下の中から自然言語を選べ
  - 英語、C言語、日本語、Python

# 前回の追加（皮肉や比喩）

- 皮肉かもしれない
  - その服おしゃれだね
  - マイペースだね
  - 丁寧なレポートですね
- 比喩であり実際には（ほぼ）あり得ない
  - ジャングルのような部屋
  - 戦場のような職場
  - 太陽のような笑顔
  - 雲のようなパンケーキ
- やっぱり自然言語は難しい

# 自然言語処理で扱う分野、技術領域

- 言語モデル/大規模言語モデル
- テキスト分類
- 情報抽出
- 情報検索
- 文章要約
- 機械翻訳
- 質問応答

# 言語モデル(1)

- n-gram
  - 文章中でn個の連続する単語の確率分布
  - n=1（unigram）：単語の出現頻度
  - n=2（bigram）：ある2つの単語が並ぶ頻度
  - n=3（trigram）：ある3つの単語が並ぶ頻度

- n-gramの限界
  - nが増えると指数関数的にデータが増える
    - しかも、ある程度以上の頻度を維持しなければ精度低下を招く
    - 頻度を増やすにはデータを増やすことが必須
    - 計算量の爆発
  - 頑張ってnを増やしたとしても、n+1以上の情報を使うことができないという限界

# 言語モデル(2)

- 指数関数モデル
  - 文脈を導入する（話題の種類、感情、時刻や場所、話者の情報など）
  - 何を文脈とするのかが難しいが、適切に選べばある程度ターゲットを絞ることができる
    - 特定の応用に特化

- ニューラルネットワーク
  - 次の単語を予測
  - 昨年度のword2vecで学習方法を説明した
    - CBOW（穴埋め）
    - skip-gram（前後予測）

# 大規模言語モデル

- Large Language Model : LLM
  - 言語モデルを大規模化したら天才になりました
    - そこに知能はあるのか？（十分あるように見える）
    - 理由はあまり解明されていない
  - 言語モデルと比較にならないほどの応用範囲（生成AIなど）
  - RNN（Recurrent Neural Network）、Transformer（詳細は深層学習で）

# （大規模じゃない）言語モデルの応用

- 予測変換
  - 確率から求まる
- 音声認識
  - 似ている単語から最もそれらしいものを選ぶ
- 入力間違いの訂正
  - 類似する確率の高い単語に置き換え
- 手書き認識
  - 似た文字の中から確率の高い文字を選ぶ

# テキスト分類

- スパム検出
  - 電子メールの時代に大活躍した
    - 従来は送信者やNGワードなどのルールベースだったが、機械学習の導入で飛躍的に精度が向上した
- トピック分類
  - ターゲティング広告配信にも使われる
- センチメント（感情）分析
  - 好意的/中立/敵対的
  - 喜怒哀楽
  - 強気/弱気
- 技法
  - ナイーブベイズ、ロジスティック回帰、SVM (Support Vector Machine)など

# 情報抽出

- トピック抽出
- 周辺情報
- 関連情報
- OCR

# その他(1)

- 情報検索
  - 大規模なデータベースからユーザの問い合わせに応じて関連情報を検索する
  - Web検索や辞書検索など
- 文章要約
  - 文章の主な事柄と全体的な意味を維持しながら、元の文章より短い要約を作成する
  - ニュースの見出し生成や議事録作成に使われる

# その他(2)

- 機械翻訳
  - 文章をある言語から別の言語に変換する
  - Google 翻訳やDeepLなど
- 質疑応答
  - 自然言語で尋ねられた質問に自動的に回答する
  - 問い合わせ用Botでも使われる
  - 自由に回答内容をカスタマイズすることが求められる

# 自然言語処理で用いる汎用ツール

- spaCy
  - Pythonオープンソースライブラリ
  - 自然言語処理システムの構築用
  - 製品用途向け
    - 速度性能重視で改造向けではない
- NLTK
  - Pythonプラットフォーム
  - 品詞付与、意味解析のような基本的な機能を提供
  - 教育研究用途向け
    - シンプルで柔軟
- CoreNLP
  - Javaオープンソースライブラリ
  - 単語分割や品詞付与
  - 教育研究用途向け
    - 高精度

# 辞書とコーパス

- 人間が言語を読み書き／話し聞きする際、言語に関する様々な知識を利用している
- （人間が用いる）国語辞典
  - 見出し語の品詞、活用形などの「形態情報」
  - 見出し語の（複数の）「意味情報」

# 単語辞書

- 自然言語処理の単語辞書では大きく「形態情報」と「意味情報」を扱う
- 形態情報
  - 読み、品詞、活用型（カ行五段など）
- 意味情報
  - コンピュータは言葉の意味を「直接は」理解できない（現在でも）
  - 古典的には格フレームと呼ばれる知識が記述される（詳細は第6 回）
    - ある単語が文中に現れるときに、どのような単語とともに現れるか
    - 記述例：はじく（「接触してきたものをはね返す」）という語義に対応する格フレーム（※IPA の計算機日本語基本動詞情報より）
    - [レインコート, 下敷き, ガラス, 羽]ガ[水, 雨, インク, ピストルの弾丸]ヲ

# シソーラス（概念階層）

- 人手で作成した階層的類義語辞書
- シソーラスでは近くの単語同士は類似しており、遠くの単語同士は類似していないと考えられる
- よって2 つの単語の類似度は共通する上位ノードの深さを基に計算できる
- 2 つの単語の類似度：$sim(a, b)$
  - $sim(a, b) = \frac{d_c \times 2}{d_a + d_b}$
  - $d_a, d_b$：a, bの単語の深さ
  - $d_c$：2 つの単語の共通の上位語の深さ

# シソーラスの例

- 日本語
  - weblio 類語辞典 https://thesaurus.weblio.jp/
  - goo 類語辞典 https://dictionary.goo.ne.jp/thsrs/
  - 言語資源開発センター分類語彙表 https://clrd.ninjal.ac.jp/goihyo.html
  - 日本語WordNet https://bond-lab.github.io/wnja/
  - 角川類語新辞典（書籍、アプリなど）
  - NTT 日本語語彙大系（書籍、CD-ROM など）
- 英語
  - weblio 英語類語 https://ejje.weblio.jp/english-thesaurus/
  - Cambridge Thesaurus https://dictionary.cambridge.org/ja/thesaurus/
  - WordNet https://wordnet.princeton.edu/
  - Oxford Learner’s Thesaurus（書籍など）

# コーパス

- 特定のテーマに沿って文章を集めたもの

- 生コーパス
  - 収集したままの、何も情報を付加していないコーパス
- タグ付けコーパス
  - 何らかの情報（品詞、構文構造、語義など）を付加したもの
- パラレルコーパス
  - 複数の言語間の対訳データを収集したコーパス
- 目的に応じて適切に選択

# 言語の統計

- コーパスを用いて対象とする言語を統計的に処理し、統計モデルを構築する考え方がある
  - 機械学習による自然言語処理も統計的なアプローチである
- n–gram
  - 隣接するn単位の共起（単位は、文字や単語など）
  - 例：英語の2 文字の連続についてコーパスを用いて以下の知見が得られた
    - ‘t’ の次は高頻度で‘h’ が続く。同様に‘he’, ‘in’, ‘an’, ‘er’ は高頻度で現れる
    - ‘q’ の次にはほぼ確実に‘u’ が続く
  - 他にも品詞タグ付コーパスを用いて、品詞を単位とした頻度および確率を求めることで、品詞の出現傾向を反映した情報が得られる可能性がある
    - 英語であれば、冠詞( “a”, “the” など) の後に名詞が現れやすい、など
    - 日本語であれば、名詞のあとに格助詞（「が」「の」「を」など）が現れやすい、など

# 文字単位のn-gramの例

- 元の文：大阪は良い天気です。
- ユニグラム：大/阪/は/良/い/天/気/で/す/。
- バイグラム：大阪/阪は/は良/良い/い天/天気/気で/です/す。
- トライグラム：大阪は/阪は良/は良い/良い天/い天気/天気で/気です/です。

# 統計演習

- n-gramを用いて文章生成を行う
  1. 1-gramを使って最も頻度の高い文字を先頭とする
  2. 先頭文字からはじまる2-gramを作り、最も頻度の高い組の2文字目を次の文字とする
  3. 次の文字を先頭文字として2から繰り返す
  - 20文字程度で良い
- 入力ファイルは30MB以上とすること

- 自作しても良いがLMSにサンプルがある
  - spaCyとか使ってません

# 演習課題（5点）

**締切：4/21 23:59**

- 実質的には任意課題と同じ
- レベル1（3点）
  - 入力ファイルの概要とファイルサイズ
  - ソースコード（LMSと同じ部分は省略可能）
  - 出力された文章（20文字以上）
- レベル2（+2点）
  - 出力された文章の考察
    - 良かったのか悪かったのか？
    - n-gramの性質を踏まえた考察

- 単語分割して単語単位のn-gramとしても良いが加点はない

# Python解説

- **defaultdict**
  - 標準の辞書はkeyが存在しないときにエラーになる
  - **d=defaultdict(int)**
    とすると存在しないときには0になる
    - **defaultdict(0)** ではない点には注意
  - **d["a"]+=1** は
  - **d["a"]=d.get("a",0)+1** と同じ
    - **get**の2つ目の引数を指定するとkeyが存在しないときにはその値を返す
- **lambda**（ラムダ）
  - **sorted**の**key=lambda x: x[1]** は値(value)でソートするという指示
    - デフォルトはkeyでソートする
  - lambdaの本当の意味は無名関数だが詳細は難しいので省略

# 次回

- 形態素解析
  - spaCy, ginza再び
- 小テスト？


  - ここがテストに出る確率が高い（満点のうち半分程度）
  - それ以外の場所も出る（残り半分）