トピックモデルによる短文の分類

by 佐田和也 (kazuya.sada@leadinge.co.jp)

目次

当初の目標

- 短文の分類をしたい
- データセットを自分で作りたい
    - Twitter から取得
- 余力があれば、ユーザごとのトピックの分布のパターンも調べてみたい
    - このトピックを発信する人は、あのトピックを発信しない、等

当初の懸念点

- 教師ラベルが無い
- 分類精度の決め方が分からない

データセットの収集

Tweepy

- Tweepy : Twitter API を扱うためのパッケージ

Twitter API を使って、検索結果 (JSON) を取得

- 検索キーワードを決める必要があった
  - q = 'です OR ます OR でした OR ました OR でしょう OR ましょう'
- ユーザごとのタイムラインも保存しておいて、余力があれば分析する

MongoDB

- PyMongo : MongoDB にアクセスするためのパッケージ
- NoQSLBooster for MongoDB : GUI ツール

Collections

- tw_samples : 検索で取得したツイート (112,284 件)
- users : 検索で取得したツイート (96,895 件)
    - このうち 1,000 件くらいを、タイムラインを取得する対象とする
- usr_tweets : 選んだユーザのタイムラインから取得したツイート (2,043,636 件)

LDA モデル

- 「ある単語を含む文書は、ある確率であるトピックを持つ」という表現
- ベイズ推定によって直接的な単語でなくても予測できる
  - 短文の分類に向いているのではないか
- 一個の文書について、複数のトピックの「構成比」を予測する
- トピック =「文書に潜んでいる主題」
  - 「経済」「エンタメ」などの「ジャンル」とはちょっと違う

gensim

- gensim : トピックモデリングのためのパッケージ
  - 辞書を作る
  - Bow 表現を作る
  - LDA モデルを作る

チューニング

指標

- Perplexity : 今回は使いません。
- Coherence : モデルを表す単語に一貫性があるか
- KL-divergence : トピック間の距離

パラメタ

- num_topics : トピックの数
- alpha : トピックの構成比がどうなっていると予測するか
    - 例 : すべてのトピックが同じ割合で出現
- no_below : 出現する文書が少ない単語をカット
- no_above : 出現する文書が多い単語をカット
- ストップワード（コーパス取得時の実装による）

成果

最初のモデルによる分類

- 訓練データをそのまま分類にかけた結果
  - topic 00
  - topic 01
  - topic 02
  - topic 03
  - topic 04
  - topic 05
  - topic 06
  - topic 07

特にトピックごとの特徴、トピック間の違いがわかりませんでした。