Colaboratory 上で完結する教材用個人化複雑単語識別器の開発

Colaboratory 上で完結する教材用個人化複雑単語識別器の開発

Developing Personalized Complex Word Identifier that Depends Only on Colaboratory for Educational Purposes

江原 遥 *
Yo EHARA

Abstract : Google Colaboratory is a Web-based Jupyter Notebook hosting service. At the time of the submission of this paper, it meets many requirements for educational purposes: it is openly available and requires only the Chrome web browser and availability of an Internet connection. This paper describes how to build a personalized complex word identifier only by using Colaboratory for educational purposes, especially for classroom education.

1. はじめに

Google Colaboratory は，Google 社が提供する，Web ブラウザ上で仮想環境にアクセスして Python 言語を起動させることのできるシステムである．投稿時点（2020 年 2 月 28 日）では，このサービスは Google アカウントさえあれば無料で使用することが可能である．特に，教室内で 10 人〜数十人程度の学生に教える際に有用な，下記の要件を満たしている．

(1) アカウントさえあれば，10 分程度の講義で，Colaboratory の画面の立ち上げまで非常にスムーズに到達できる．

(2) 無料で利用できる．

(3) インターネット接続環境と Chrome ブラウザさえ動作すればよいので，特段高性能なスペックを要求しない．

(4) 仮想環境であるため，機械的な故障などで 1 台だけ異なる動作をする，といったことがほとんどない．

ただし，下記のようなデメリットがある．

(1) 毎回，仮想マシンのインスタンスは基本的に初期化された状態で起動するため，大規模なデータなどをあらかじめ置いておくことは難しい．

(2) 仮想マシン環境を継続して動作させられる時間が限られている．

(2) の点については，インスタンスの継続動作時間が 12 時間であることから，講義中に途切れるなどの問題はなく，講義で利用するためだけであれば特段の問題はない．一方，(1) の点については問題がある．(1) は，機械学習を用いた課題を行う場合には，訓練データを用意する方法に工夫が必要であることを示している．

(1) の訓練データを用意する方法として，Google Colaboratory では，大容量の訓練データを用意する方法が用意されてはいるが，この方法では，教材として用いるには不便な点があることを説明する．例えば，Google Colaboratory では，クラウドサービスとして有名な Google ドライブにマウントする機能を用意しているので，この機能を使って，訓練データなど大規模なデータをあらかじめアップロードしておき，Colaboratory から使用する方法が考えられる．しかし，この方法では，講義中は，講義を受ける側が各々自分のアカウントでログインしていることが多いので，Google ドライブのマウントにかかる承認手続きが煩雑になり，時間がとられることが予想される．また，Google ドライブ上の共有設定を気を付けていないと，学生が講義をする側の見るべきではないファイルを見てしまうなどの問題も生じる．より簡便な方法は，各仮想マシンのインスタンス上で，公開されている訓練データをダウンロードする方法であろう．

本稿では，大容量の訓練データを必要とせず，学生が数十分の短時間で学習データを追加できるタスク（課題）を教材に用いることで，(1) の問題を解決する．具体的には，「個人化単語推定」のという課題を与える．個人化複雑単語推定は，文献 [9] や文献 [10] に挙げるコーパスの単語頻度を利用すれば，かなり高い精度で行えることが分かっている [14]．もちろん，より複雑なモデルを構成したり，特徴量を入れたりすることも可能ではあるが，単語頻度特徴量だけを用いれば，訓練データとしては数百 KB〜数 MB のデータで十分である．そのため，機械翻

2020 年 5 月 7 日受理
* 情報学部　コンピュータシステム学科

訳など，他の大規模なテキストデータを扱う自然言語処理のタスクと比べれば，訓練データ量は少量で済む．

本稿では，単純な機械学習の 2 値識別器を構成する問題に帰着される個人化複雑単語推定の問題について，Google Colaboratory 上だけで完結する 2 値識別器の構成法について論じる．ここで,「完結する」の意味については，Google Colaboratory は，そもそもインターネット環境で動作するシステムであるので，インターネット上で公開しているファイルを仮想マシンのインスタンス上にダウンロードして利用する場合も「完結する」と表現することにする．Google Drive など，あらかじめ訓練データ等を用意しておいたクラウド環境とマウントしないと使えない場合は,「完結する」と表現しないことにする．インターネット上で公開しているファイルだけを用いるようにしておけば，ローカル環境など，Colaboratory 以外の環境に移植することも容易になる．

本稿の貢献は，前述の (1) に挙げた訓練データの用意の問題を解決し，学生が訓練データを講義時間内の短い時間で作成しても，十分な精度で動作するタスクとして，個人化複雑単語推定のタスクを見出したこと，および，その教材作成例を，実例をもって示したことにある．具体的には，機械翻訳など大規模なコーパスを必要とする他のタスクと比較して，機械学習による 2 値識別器を少量の公開されているデータセットから構成する個人化複雑単語推定のタスクが取り扱いやすいことを，実例をもって示した．

2. 関連研究

自然言語処理分野，機械学習分野では，Colaboratory を用いた教材開発は盛んである．特に，自然言語処理分野においては，機械翻訳（自動翻訳）[5] などのタスクでは，多くの Colaboratory を用いた教材が作られている．こうした教材では，訓練データのために数百 MB にのぼるコーパスをダウンロードしないといけないことが多い．仮想マシン上へのダウンロードであるため，仮想マシンに接続しているローカルマシン上では特にスペックは必要がないものの，接続が不安定になることも多い．このため，数百 MB 程度以上の訓練データをダウンロードしないといけないようなタスクは，機械学習の教材としては避けたいのが実情であろう．また，機械翻訳に用いられる手法は日進月歩であり，最新の手法を教えようと思うと，頻繁に更新する必要がある．最新の手法は先進的すぎるため，基礎的な機械学習の講義とつなげて教えることも難しい.「Google 翻訳」などのサービスでは，最新の手法を大量の対訳コーパス（分単位の対訳対データ）から実装したものが，無料で公開されているため，普段，そうしたサービスを使い慣れている学生は，逆に，こうした最新手法を使った手法の精度が低く見えてしまうこともあり得る．

個人化複雑単語推定は，単語テストに対して各学習者が正答/誤答した，というデータから，単語テスト中にない単語について，単語テストを受けた学習者が，所与の単語を知っているかどうかを判定する問題である．機械学習の用語で言えば，Support Vector Machines [8] やロジスティック回帰といった，基礎的な手法で実装可能なタスクである．このため，機械学習の初歩の講義と結び付けて教えることが容易である．複雑単語推定自体は，自然言語処理分野でメジャーとまではいえないものの，一応，Shared Task が行われている程度には，有用性が認められているタスクである [16]．学生が，単語帳を自分で作って経験があれば，有用性が理解できる利点もある．

表 1 に，複雑単語推定を他の機械学習タスクと比較した表を載せる．表のように，個人化複雑単語推定以外のタスクは，少々前の研究でも複雑な機械学習モデルやニューラルネットワークが使われており，専門的すぎて機械学習の初歩と関連付けて学生に教えることが難しいことがわかる（表 1）．

3. 開発

まず，COCA コーパス，ならびに BNC コーパス上の単語頻度を読み込む図 1, 図 2．これらは，新聞記事など大抵の英文上で現れる単語頻度を網羅しても，無圧縮状態で，高々数 MB のサイズで済む．こうした単語頻度情報は，もともとのコーパスのテキスト情報そのままではなく，統計情報（モデル）であるので,「著作権法４７条の７」の規定により，著作者の許可を得ずに，非営利目的に限定せずに利用者に利用してもらうことが可能であると考えられる [3]．そこで，ここでは，あらかじめ作成した単語頻度情報を，URL の形で提供して，各々の仮想マシンのインスタンスから読み込ませている．

次に，文献 [11] で公開している，単語テストのデータセットを読み込む．これは，100 人に 100 語の単語テストを解いてもらったデータセットである．これを用いることで，新しく学習者を 1 人〜1 グループ分数名追加したとしても，他の学習者のデータを参照することで，識別が安定した精度で出るようになる．図 3 でダウンロード，図 4 で読み込みを行っている．

文献 [11] のデータセットでもとにしている Vocabulary Size Test [15] の回答は，PDF で公開されている．回答をどこかに別にアップロードすることも技術的には可能であろうが，回答データは研究成果として公開されているものであるため，直接 1 次情報の PDF を読み込んでいる（図 5）．PDF をテキストとして読み込むためには，Python 上の pdfminer.six というライブラリを利用した [2]．

最後に，ダウンロードした単語頻度情報から単語の特徴量ベクトルを構成し（図 6）, 訓練データとテストデー

表 1: 複雑単語推定タスクの他の機械学習タスクとの比較.

タスク名	現状の典型的な機械学習問題	無料入手可能な訓練データサイズ
機械翻訳	LSTM, 注意機構 [7]	74MByte [6]
感情分析	系列ラベリング [17]	80MByte[4]
個人化複雑単語推定 [13]	2 値判別	5MByte 未満

```
!wget -O coca_w1.txt 'https://www.dropbox.com/s/8owrzc57y7ush0l/coca_w1.txt?dl=0'
coca_fdl = [(line.strip().split()[1],int(line.strip().split()[0])) for line in open('coca_w1.txt', r'r')]
import nltk
from nltk import FreqDist
coca_fd=FreqDist(dict(coca_fdl))
coca_pd=nltk.LaplaceProbDist(coca_fd)
```

図 1: COCA コーパスの読み込み.

```
#British National Corpusの単語頻度表をダウンロードする
!wget -O bnc_freqdist.pickle 'https://www.dropbox.com/s/lcvyfqk2ezlalks/bnc_freqdist.pickle?dl=0'
```

図 2: BNC コーパスの読み込み.

```
!wget http://yoehara.com/download/293/?uid=58af854d3a  -O ehara2018dataset.zip

--2020-02-28 07:49:43--  http://yoehara.com/download/293/?uid=58af854d3a
Resolving yoehara.com (yoehara.com)... 59.106.27.175
Connecting to yoehara.com (yoehara.com)|59.106.27.175|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6344 (6.2K) [application/zip]
Saving to: ‘ehara2018dataset.zip’

ehara2018dataset.zi 100%[===================>]   6.20K  39.2KB/s    in 0.2s

2020-02-28 07:49:45 (39.2 KB/s) - ‘ehara2018dataset.zip’ saved [6344/6344]
```

図 3: 文献 [11] データセットのダウンロード.

```
import csv
import codecs
VSTres_raw = {}
with codecs.open('EVKD_lrec2018.csv', 'r', 'utf-8', 'ignore') as f:
    reader = csv.reader(f)
    for j,row in enumerate(reader):
        if j==0:
          continue
        VSTres_raw[j-1] = [] + row[5:105]
```

図 4: 文献 [11] データセットの読み込み.

```
#単語テストの正解ファイルはPDFでしか提供されていないので読み込む
!wget https://www.victoria.ac.nz/lals/about/staff/publications/paul-nation/VST-version-A_answers.pdf
```

図 5: VST の正解データの読み込み.

タに分けて 2 値判別を行う（図 7）.

100 語のうち 90 語を訓練，10 語をテストデータに使ったところ，精度は 0.7772 であった．これは，文献 [12] などと比較しても，良い精度と言える.

作成したソースコードは文献 [1] に示す URL から入手可能である.

```
def get_wordfeat(lemma):
    return [- bnc_pd.logprob(lemma), - coca_pd.logprob(lemma)]
```

図 6: 単語に対する特徴量ベクトルの構成.

```
for learnerid in learnerids:
  wordfeats_train = mk_feats([VSTwords[wordid] for wordid in trainwordids])
  learnerfeat_train = np.repeat(mk_learnerfeat(maxlearner, learnerid)[np.newaxis,:], len(trainwordids),axis=0)
  X_trains.append( np.hstack((learnerfeat_train,wordfeats_train)))
  y_trains.append( np.array(VSTres[learnerid],dtype=np.int)[trainwordids])

  wordfeats_test = mk_feats([VSTwords[wordid] for wordid in testwordids])
  learnerfeat_test = np.repeat(mk_learnerfeat(maxlearner, learnerid)[np.newaxis,:], len(testwordids),axis=0)
  X_tests.append( np.hstack((learnerfeat_test,wordfeats_test)))
  y_tests.append( np.array(VSTres[learnerid],dtype=np.int)[testwordids])
```

図 7: 訓練データ・テストデータ分割.

```
clf = GridSearchCV(LogisticRegression(),
                                param_grid={'C':np.logspace(-1,1,5)},
                                cv=10
                                )
```

図 8: 2 値識別器の構成.

4. おわりに

本稿では，Colaboratory を用いた機械学習の教材を準備する際の問題点として，訓練データの準備の問題を取り上げた．訓練データが少量であり，さらに，学生が講義時間内で訓練データを追加することも可能なタスクが望ましい．このようなタスクとして，自然言語処理分野における個人化複雑単語推定が適していることを示し，実際に，Colaboratory 上で完結するスクリプトの開発について報告した．個人化複雑単語推定は，必要とする訓練データも小さい割には，単純な機械学習の 2 値判別の問題に帰着することができる利点がある．今後の課題としては，2 値判別など機械学習の基本的なタスクに帰着できる有用な応用の例を増やすことが挙げられる.

謝辞

本研究は静岡理工科大学 2019 年度提案型教育研究費研究プロジェクト B の支援を受けたものである.

参考文献

(1) http://pr1.work/0/kiyou.
(2) https://arakan-pgm-ai.hatenablog.com/entry/2018/01/07/080000.
(3) https://storialaw.jp/blog/4936.
(4) https://www.tensorflow.org/datasets/catalog/imdb_reviews.
(5) http://www.second-academy.com/lecture/SHU16982.html.
(6) https://alaginrc.nict.go.jp/WikiCorpus/download.
(7) Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. Weighted transformer network for machine translation. *arXiv preprint arXiv:1711.02132*, 2017.
(8) Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, Vol. 2, No. 3, pp. 1–27, 2011.
(9) The BNC Consortium. *The British National Corpus, version 3 (BNC XML Edition)*. Bodleian Libraries, University of Oxford, 2007.
(10) Mark Davies. The 385+ million word corpus of contemporary american english (1990–2008+): Design, architecture, and linguistic insights. *International journal of corpus linguistics*, Vol. 14, No. 2, pp. 159–190, 2009.
(11) Yo Ehara. Building an english vocabulary knowledge dataset of japanese english-as-a-second-language learners using crowdsourcing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
(12) Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. Mining words in the minds of second language learners: learner-specific word difficulty. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, Mumbai, India, December 2012.
(13) Tomoyuki Kajiwara and Mamoru Komachi. Complex word identification based on frequency in a learner corpus. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 195–199, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
(14) John Lee and Chak Yan Yeung. Personalizing lexical simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 224–232, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.
(15) Paul Nation and David Beglar. A vocabulary size test. Vol. 31, No. 7, pp. 9–13, 2007.
(16) Gustavo Paetzold and Lucia Specia. Benchmarking lexical simplification systems. In *LREC*, 2016.
(17) Andrea Vanzo, Danilo Croce, and Roberto Basili. A context-based model for sentiment analysis in twitter. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers*, pp. 2345–2354, 2014.