§4 Заключительный этап: командная часть

Задача 4.1

Описание предметной области.

Выявление информации из текста - одна из самых сложных и интересных задач машинного обучения. Речь идет не о разборе предложения или структуре текстов, а о таких вопросах как определения авторства или стиля. Говоря о машинном обучении неизбежно встает вопрос о том, как программе обрабатывать и находить новую информацию в огромных объемах данных, которые даже человек не может осмыслить. Например, нет такого в мире специалиста, который бы мог точно определить авторство любой заметки из 200 авторов начала 20го века. Возможность обучения без учителя, когда вручную отмечаются правильные ответы лишь на небольшом количестве данных - одна из самых актуальных задач машинного обучения. Ведь как только алгоритмы смогут эффективно обучаться на больших объемах без предварительной ручной разметки создание сильного искусственного интеллекта будет сильно ближе.

Описание актуальной задачи.

- В качестве данных предлагается анализировать личные дневники 19го и начала 20го века. Настоящее авторство текстов - популярный вопрос среди искусствоведов, дискуссии о котором не утихают десятилетиями. Благодаря аналитике больших данных можно не просто сказать к чьему перу относится запись, но и выявить под чей стиль она подходит и выявить важные факты, передав искусствоведам эффективный инструмент анализа. Однако, возможен случай когда у набора записей не осталось сведения об авторах, и список авторов неограничен. Чтобы разобрать эти записи можно применять кластеризацию, и находить схожие заметки. При этом, в обучающей выборке лишь небольшое количество размеченных записей (около 2000), и большое (десятки тысяч) неразмеченных данных. Необходимо реализовать обучение с частичным привлечением учителя (semi-supervised learning).
- Всего на сайте около 200 авторов, и 100 000 записей дневников.
- В качестве обучающей выборки представлено 100 000 записей, из которых около 2000 будет отмечено авторство.
- Тестовая (полностью размеченная) выборка https://www.dropbox.com/s/06b576ea8f2795w/test_1.csv
- Обучающая (не полностью размеченная) выборка https://www.dropbox.com/s/isfberyv2bcu9gn/train_1.csv
- Пример ответа https://www.dropbox.com/s/c1mvyexhtst13zj/resExample_1.csv
- Есть неограниченное количество попыток

Код проверки и генерации правильного решения:

```
import csv
import random
import requests

def generate():
    return ""
url =
'https://stepik.org/media/attachments/lesson/43477/test_full_1.csv'

def solve(dataset):
```

```
    request = requests.get(url)
    csvMy = request.text
    return csvMy

def check(reply, clue):
    orig_dict = dict()
    csvMy = clue.splitlines()

    for line in csvMy[1:]:
       # print(line)
        splitedLine = line.split(",")
       # print(splitedLine)
        orig_dict[int(splitedLine[0])] = int(splitedLine[1])
    replyS = reply.splitlines()
    reply_dict = dict()

    for line in replyS[1:]:
       # print(line)
        splitedLine = line.split(",")
      #  print(splitedLine)
        reply_dict[int(splitedLine[0])] = int(splitedLine[1])
    trueCount = 0
    for key, value in reply_dict.items():
        if (orig_dict[key] == int(value)):
            trueCount+=1

    res = trueCount/float(max(len(orig_dict),len(reply_dict)))

    if res >=1:
        return True
    #if (f1_macro(confusion_matrix(model, original)) == 1):
    #    return True
    #balls = f1_macro(confusion_matrix(model, original))
    balls = 100*res
    resT = "В первой задаче набрали "+str(balls)+" баллов"

    return res, resT
```

Задача 4.2

Выделение из текста заранее известных сущностей - одна из самых распространенных

Середина двадцатого века один из самых сложных периодов нашей страны. По разным причинам (на войне, или в результате репрессий) пропало множество людей, о которых практически не осталось данных. Все данные которые мы можем выяснить это скупые карточки и журнальные записи, оставленные в разных формах. Множество общественных организаций занимается оцифровкой этих данных, чтобы наши соотечественники могли узнать судьбу своих родных. Однако, как писалось ранее, эти данные представляют собой скупые и не структурированные заметки из разных журналов и форм, для того чтобы люди могли найти своих родственников необходимо из этих заметок выделить как можно больше классифицированной информации (сущностей)

Необходимо осуществить выделение сущностей из коротких текстов карточек. При этом не все записи содержат все сущности. Задача называется named entity recognition.

Список сущностей:

Имя, фамилия, отчество, "никнейм или кличка", пол, год рождения, место рождения, работы, место проживания, был ли арестован, день ареста, месяц ареста, год ареста, кем арестован, день суда, месяц суда, год суда, статья обвинения, наказание, был-ли расстрелян, день расстрела, месяц расстрела, год расстрела, день реабилитации, месяц реабилитации, год реабилитации, источник, были ли дети, сам текст.

В качестве обучающей выборки представлены уже разобранные варианты.

Датасет:
https://www.dropbox.com/s/cofmb51avs7ip42/train2.csv?dl=0
или
https://drive.google.com/file/d/0B2coL5p0DNtcR2pURlVxUXVQZHM/view?usp=sharing

Типовое решение:
https://www.dropbox.com/s/j3pi61xka3f8fe2/Task2_Baseline1.py?dl=0

Формат выдачи:

```
,id,text
43353,116372,"Семенов Андрей Васильевич, 1916 г.р., место рожд.:
Тетюшский р-н, д.Чувашский Чикилдым, жил там же. Чуваш, пред., к-з
""Ленин-солете"". Арестован 15.11.42, осужден 22.3.43 по ст. 58-14.
Приговор: 10 лет лишения свободы, поражен. прав на 2 года."
27477,104180,"Кунгурцев Ефим Ипполитович (Ипатович), 1906 г.р., место
рожд.: Тюменская обл., Упоровский р-н, д.Москва (Млоква), жил:
г.Казань. Русский, 3 детей, электромонтер, грузчик, кожз-д им.Ленина.
Арестован 23.10.41 (""занимался политическим бандитизмом, участник к/р
группировки, недовольство политикой Сов.вл.""), осужден Особым
совещанием НКВД СССР 20.6.42 по ст. 58-10 ч.2. Приговор: 10 лет ИТЛ.
Реабилитирован 29.5.89."
42127,115014,"Саитов Шагивалей, 1885 г.р., место рожд.: Апастовский р-
н, с.Эбалаково, жил там же. Татарин, бедняк. Арестован 11.10.29
(""участник религиозных выступлений""). Предъявлено обвинение по ст.
58-10. ГПУ ТАССР 31.12.29 дело прекращено за недоказанностью
обвинения."
```

Формат сдачи:

```
,id,lastname,firstname,middlename,alternative_name,sex,birthyear,birth
_place,job,liveplace,is_arested,arested_day,arested_month,arested_year
,state_authority,judgment_day,judgment_month,judgment_year,prosecution
_article,prosecution,is_shooted,shooted_day,shooted_month,shooted_year
,rehabilitation_day,rehabilitation_month,rehabilitation_year,source,ha
d_children
51866,203290,Шадров,Тимофей,Григорьевич,,м,1874,"Балтасинский р-н,
с.Средний Кушкет",крестьянин.,"Балтасинский р-н, с.Средний
Кушкет",Арестован,23.0,10.0,1930.0,тройкой ГПУ
ТАССР,8.0,12.0,1930.0,"58-8, 58-10. (""бывший торговец, агитация
против хлебозаготовок, угрозы активу"")",5 лет
концлагерей,,,,,23.0,5.0,1989.0,КП Республики Татарстан,
26116,102682,Корноухов,Родион,Демьянович,,м,1887,"Альметьевский р-н,
с.Новая Елань","колхозник, колхоз ""Красная Елань""."","Альметьевский
р-н, с.Новая Елань",Арестован,11.0,5.0,1944.0,,,,,"58-10 ч.2.
(""профашистская агитация"")",Покончил жизнь самоубийством
_12.05.1944_ в КПЗ Акташского РО НКВД ТАССР,,,,,5.0,11.0,1996.0,КП
Республики Татарстан,
51009,202338,Чалкин,Василий,Иванович,,м,1910,"Алексеевский р-н,
с.Лебедино","колхозник, бригадир.","Алексеевский р-н,
с.Лебедино",Арестован,13.0,9.0,1938.0,Верховным судом
ТАССР,24.0,5.0,1939.0,"58-2, 58-7, 58-8, 58-11. (""участник эсеровской
```

```
террор. подрывной организации"")",оправдан,,,,,,,,КП Республики
Татарстан,
```

Оба формата легко читаются и пишутся с помощью csv.DictWriter с дефолтными настройками.
Порядок строк сохранять не обязательно, они проверяются по полю " (первый идентификатор), не теряйте его.

Про уточнения, которые мы сделали:

- В prosecution везде убрали точки в конце
- В birth_place, если там было строки вида "(место жительства не определено)", заменили их на пустую строку ""
- В alternative_name отформатировали корректно имена. Они должны быть отсортированы без пробелов и соединены через набор символов "| |", например "Лабаева| |Лазарева"
- Есть неограниченное число попыток.

Проверка и генерация верного решения:

```
# -*- coding: utf-8 -*-

import random
import io
import csv
import urllib.request

TEST_HEADERS = ",id,text"
ANSWER_HEADERS =
",id,lastname,firstname,middlename,alternative_name,sex,birthyear,birt
h_place,job,liveplace,is_arested,arested_day,arested_month,arested_yea
r,state_authority,judgment_day,judgment_month,judgment_year,prosecutio
n_article,prosecution,is_shooted,shooted_day,shooted_month,shooted_yea
r,rehabilitation_day,rehabilitation_month,rehabilitation_year,source,h
ad_children"
COMPARSION_HEADERS =
"lastname,firstname,middlename,alternative_name,sex,birthyear,birth_pl
ace,job,liveplace,is_arested,arested_day,arested_month,arested_year,st
ate_authority,judgment_day,judgment_month,judgment_year,prosecution_ar
ticle,prosecution,is_shooted,shooted_day,shooted_month,shooted_year,re
habilitation_day,rehabilitation_month,rehabilitation_year,source,had_c
hildren"

DATA_PUBLISH_COUNT = 3000

LOCAL = False
VERBOSE = False

DOWNLOAD_URLS = {
    'test_clear':
"https://stepik.org/media/attachments/lesson/43673/test_clear.csv",
    'test_full_clear':
"https://stepik.org/media/attachments/lesson/43673/test_full_clear.csv
"
}

def log(*args):
```

```
    if VERBOSE:
        print(*args)

def get_source(name):
    if LOCAL:
        return parse_file_source(name + '.csv')
    else:
        log("Downloading {} from {} ".format(name,
DOWNLOAD_URLS[name]))
        return parse_lines(io.StringIO(download(DOWNLOAD_URLS[name])))

def download(url):
    request = urllib.request.Request(url)
    response = urllib.request.urlopen(request)
    return response.read().decode('utf-8')


def parse_lines(lines):
    reader = csv.DictReader(lines)
    res = {}
    for row in reader:
        res[row['']] = row
    return res

def parse_file_source(path):
    with open(path) as csvfile:
        return parse_lines(csvfile)

def write_output(data, fieldnames):
    output = io.StringIO()
    writer = csv.DictWriter(output, fieldnames=fieldnames)

    writer.writeheader()
    for value in data:
        writer.writerow(value)

    return output.getvalue()

def calculate_row_hits(original_value, model_value,
comparsion_headers):
    return sum([int(original_value[h].lower().strip() ==
model_value[h].lower().strip()) for h in comparsion_headers])

def generate():
    data = get_source('test_clear')
    samples = random.sample(list(data.values()), DATA_PUBLISH_COUNT)
    samples.sort(key=lambda x: x[''])
    return write_output(samples, TEST_HEADERS.split(','))

def solve(dataset):
    sample_texts = parse_lines(io.StringIO(dataset))
    answers = get_source('test_full_clear')
    sample_answers = [answers[key] for key in sample_texts.keys()]
    sample_answers.sort(key=lambda x: x[''])
    return write_output(sample_answers, ANSWER_HEADERS.split(','))

def check(reply, clue):
    try:
```

```
        model_data = parse_lines(io.StringIO(reply))
        original_data = parse_lines(io.StringIO(clue))
    except Exception as e:
        return False, "Не удалось распарсить текст"

    for key in original_data.keys():
        if key not in model_data:
            return False, "Отсутствует запись {}".format(key)

    dataset_size = max(len(model_data), len(original_data))

    comparsion_headers = COMPARSION_HEADERS.split(',')
    max_hits = len(comparsion_headers)

    total_max_hits = max_hits * dataset_size
    total_row_hits = 0

    log("Total recs: {}, total values: {}".format(dataset_size,
total_max_hits))
    for (key, value) in original_data.items():
        try:
            row_hits = calculate_row_hits(value, model_data[key],
comparsion_headers)
        except KeyError as e:
            return False, "Отсутствует поле " + str(e)

        total_row_hits += row_hits
        log("{} of {} hits for row {}".format(row_hits, max_hits,
key))

    acc = total_row_hits / total_max_hits
    balls = 100*acc
    comment = "Вы получили за эту задачу "+str(balls)+" баллов"
    return acc, comment


#sample_data = generate()
#log(sample_data)
#print(check(solve(sample_data), solve(sample_data)))
```