Практическое задание по курсу "Обработка текстов". Осень 2015

Практическое задание включает решение двух задач:

- Выявление личных оскорблений в дискуссиях пользователей Livejournal.
- Определение демографических атрибутов пользователей (образование и политические взгляды) по их сообщениям.

Данный документ содержит описание первой задачи и общих требований. Описание второй задачи появится позднее.

Постановка задачи

Целью задания является создание системы, позволяющей выявлять оскорбления в русскоязычных комментариях пользователей livejournal.

Система должна

- выявлять оскорбления в адрес других участников дискуссии,
- определять сообщения, не содержащие оскорбления и оскорбления не в адрес участников дискуссии как не оскорбления.

Пример:

> *Ну судя по тому, какую чушь ты пишешь у тебя должно быть родовая травма головы была...*
>
> Ответ системы: Оскорбление

Пример:

> *Кличко-младший - [censored]. Это факты, а не фантазии.*
>
> Ответ системы: Не оскорбление (т.к. не затрагивает участников дискуссии)

Пример:

> *Отличные фотографии, спасибо.*
>
> Ответ системы: Не оскорбление

Решение задачи

Практические аспекты

Решения должны быть написаны на языке Python 3.5. Можно использовать все стандартные библиотеки, а также

- NLTK - инструменты для обработки текстов

- scikit-learn - алгоритмы машинного обучения
- numpy - работа с многомерными массивами

Доступ в Интернет на проверяющей машине закрыт. По требованию может быть предоставлен доступ к https://api.ispras.ru

Теоретические аспекты

Предполагается использование алгоритмов машинного обучения. Для обучения алгоритма требуется придумать признаки и дать ему на вход правильные примеры - обучающий корпус.

Разметка обучающего корпуса

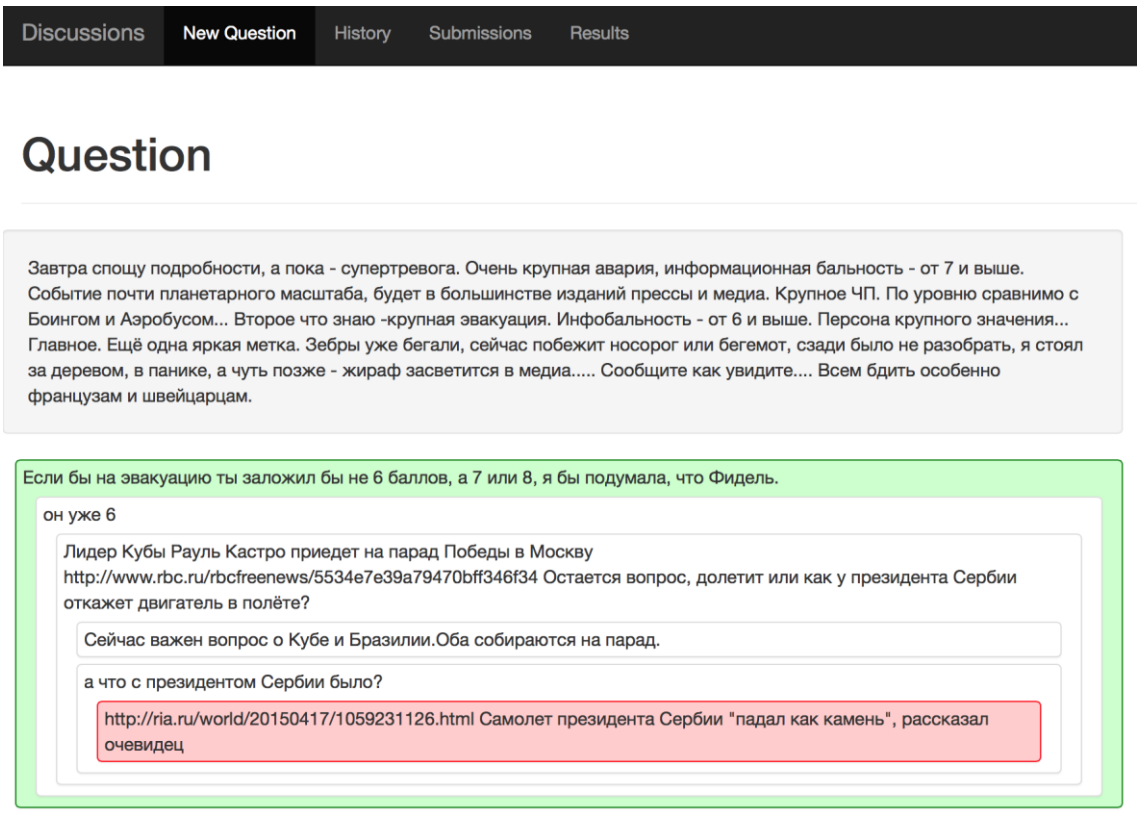
Считается, что чем больше обучающий корпус, тем лучше работает алгоритм. Однако создание большого обучающего корпуса - довольно трудоемкая задача, непосильная одному человеку. Поэтому предлагается создать его с помощью коллективной работы.

Для определения оскорблений полезно знать контекст, поэтому предполагается размечать дискуссии целиком (дискуссия состоит из блог-поста и комментариев к нему).

Чтобы облегчить работу по разметке сообщений был сделан сайт: http://discussions.at.ispras.ru. Каждому из вас предстоит разметить 30 дискуссий.

Для разметки корпуса необходимо зарегистрироваться на сайте http://discussions.at.ispras.ru. Пожалуйста, вводите правильные личные данные, так как они будут использоваться при выставлении зачетов. *Вне рамок практикума эти данные использоваться не будут.*

После регистрации появится окно разметки.



При нажатии на комментарий (за исключением первого поста) он меняет цвет.

- Белый - не размечено (или нет уверенности в типе)
- Зеленый - не оскорбление
- Красный - оскорбление

В дискуссии вам необходимо отметить красным комментарии — оскорбления в адрес участников дискуссии и зелёным комментарии, не являющиеся оскорблениями в адрес участников дискуссии. Для повышения надёжности разметки каждая дискуссия будет размечена более чем одним чем одним человеком.

Для того, чтобы сделать разметку более однозначной и сделать более однозначным понятие оскорбление был создан **манифест** по разметке.

После окончания разметки поста, необходимо нажать кнопку "Submit".

Во вкладке "History" можно посмотреть размеченные посты и скорректировать разметку.

Тренировочный корпус

Тренировочный корпус будет доступен для скачивания в формате json. Для извлечение информации из этого файла рекомендуется использовать стандартную библиотеку Python с одноименным названием.

Для синхронизации обучения и тестирования в течении недели, корпус будет состоять из дискуссий, размеченных автором классификатора, плюс все дискуссии, размеченные в течении предшествующей недели.

Кроме того, возможно использование любых внешних данных для обучения. Об использовании таких данных необходимо сообщить письмом и прислать их вместе с решением.

Тестирование

Вместе с кнопкой скачивания тренировочного корпуса появится ссылка на форму для загрузки файла и личную страницу со статистикой. На личной странице находится статистика со всеми результатами в т.ч. результатами последнего тестирования (дата, описание, достоверность).

Загрузка решения. Загружаемый файл должен представлять собой zip архив с любым именем. Архив должен обязательно содержать:

- классификатор в файле InsultDetector.py. В файле должен содержаться класс InsultDetector. В классе должны присутствовать методы
    - train(self, discussions). На вход метод train получает список размеченных дискуссий (см описание формата дискуссии). Метод train ничего не возвращает. Внимание: метод train будет вызываться отдельно, так что не стоит вызывать его в конструкторе класса.
    - classify (self, discussions), который получает на вход список неразмеченных дискуссий. Метод classify должен для каждого сообщения в дискуссии (кроме корня)
    - определить является ли сообщение оскорблением. Метод classify должен вернуть список размеченных дискуссий. Полученные на вход дискуссии можно изменять.
- (Пустой) файл __init__.py в корне архива. (Требования к пакетам Python).
- Описание применяемых алгоритмов в файле description.txt

- Все файлы должны быть в кодировке UTF-8
- все используемые внешние библиотеки, кроме библиотек пакетов NLTK, scikit-learn и numpy (они доступны автоматически).

Описание дискуссии

Дискуссия представленна в виде dictionary вида

{"root": корень дискуссии}.

Корень дискуссии в свою очередь является dictionary вида

```
{
        "id": уникальная для каждого сообщения строка,
        "text": текст сообщения,
        "children": список ответов на сообщение
}
```

В свою очередь children это тоже dictionary вида

```
{
        "id": уникальная для каждого сообщения строка,
        "text": текст сообщения,
        "insult": True если сообщение является оскорблением, False инач,
        "children": список ответов на сообщение (если на сообщение были ответы)
}
```

Проверка решения

Результаты тестирования появятся на личной странице, как только закончится обучение и тестирование. При загрузке нового классификатора обучение будет производится на корпусе, размеченном автором классификатора, плюс все дискуссии, размеченные в течении предшествующей загрузке недели.

В течении недели студенты не видят прогресс своих коллег и могут посмотреть только свой результат. В конце каждой недели (каждый вторник в 23.59.59) будет производится переобучение последнего присланного решения от каждого студента на новом корпусе, а результаты тестирования будут показаны в сводной таблице.

Ограничения

1. каждую неделю можно послать только 10 версий программы (**внимание! Итоговое тестирование будет проводится на последнем загруженном решении**)
2. размер архива не может превышать 15Мб

В связи с первым ограничением, для тестирования на локальной машине рекомендуется использовать метод перекрестной проверки (http://en.wikipedia.org/wiki/Cross-validation_(statistics)). В библиотеке scikit-learn есть функции, которые могут помочь в использовании этого метода (например, KFold()).

Оценка качества

Для оценки качества используются F1-мера, в качестве положительного класса выбран класс insult.

Описание в документации к библиотеке scikit-learn: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html#sklearn.metrics.f1_score

Baseline

Baseline 1. В качестве классификатора используется наивный байесовский классификатор. В качестве признаков используются униграммы слов.

Baseline 2. В качестве второй, более сложной нижней границы используется один из стандартных алгоритмов классификации с N-граммами в качестве признаков.

Оба классификатора будут тренироваться на том же корпусе, что и присланные алгоритмы, а так же на специальном корпусе. Достоверность будет меняться соответственно.

Подсчет очков

Как было сказано выше, в конце каждой недели вы сможете посмотреть, насколько хороший классификатор вы сделали по сравнению с другими предложенными решениями. Эти результаты нужны только для понимания текущей ситуации.

В течении семестра будет три дедлайна, когда текущие результаты преобразуются в очки, которые повлияют на итоговую оценку за курс.

Расписание дедлайнов:

1. 4 ноября (учитываются все решения, присланные до 23:59:59 3 ноября)
2. 9 декабря

При наступлении дедлайнов, так же как и в конце обычной недели производится обучение и тестирование всех присланных решений. Далее производится ранжирование результатов (по F_1 мере) и начисляются очки: за 1 место – 10 очков, 2-9 и т.д. Все программы выше лучшего baseline получают по 2 очка, выше худшего - по одному очку. После этого результаты становятся доступны всем на главной странице.

Первое задание можно сдавать до второго дедлайна, однако количество полученных очков уменьшается в два раза с округлением в большую сторону. За задание выставляется максимальный из полученных баллов.

Второе задание будет выдано в начале ноября. Поэтому актуален только второй дедлайн.

Выставление оценок

После 8 декабря будут выставляться итоговые оценки.

- Для получения отметки "**Отлично**" - необходимо набрать минимум 2 балла за каждое задание и не менее 5 баллов в сумме (решения лучше baseline 2 и хотя бы раз (вовремя) попали в top-8).
- "**Хорошо**" ставится за 3-4 балла, минимум 1 балл за задание (надо вовремя побить baseline 2 для одного из заданий и baseline 1 для другого).
- Для получения отметки "**Удовлетворительно**" необходимо набрать минимум по 1 баллу за задание (побить baseline 1 для обоих заданий).
- Оценка "**Неудовлетворительно**" ставится, если хотя бы одно задание на сдано.

Внимание! Оценку "неудовлетворительно" изменить нельзя никаким образом!

Экзамен

Экзамен будет проходить во второй половине декабря. Оценка за практикум не влияет на оценку за экзамен, за исключением оценки "неудовлетворительно". **Не сдавшие практикум к экзамену не допускаются!** (Можете считать, что есть ступенчатая функция от оценки за практикум, которая в сумме с оценкой за экзамен дает либо саму оценку (если практикум сдан), либо "неудовлетворительно", если практикум не сдан).

Для студентов ФКН ВШЭ

Итоговая оценка за курс (по 10-бальной шкале) является суммой оценок (по 5-бальной шкале) за практическую часть и за экзамен.

Оценка "неудовлетворительно" за любую часть является блокирующей, то есть итоговая оценка тоже будет "неудовлетворительно".

Дополнительные вопросы

- Все вопросы, кроме технических, задавайте на сайте http://tpc.at.ispras.ru, либо пишите на turdakov@ispras.ru
- Все технические вопросы относительно проверки заданий просьба присылать на laguta@ispras.ru либо спрашивать в разделе сайта, посвященном практикуму.
- Для установки внешних модулей (NLTK, scikit-learn, BeautifulSoup) рекомендуется использовать easy_install из пакета setuptools.

Вспомогательная литература

- Тоби Сегаран, “Программируем коллективный разум” (Книга про прикладное применение некоторых технологий искусственного интеллекта, включая машинное обучение, в Web 2.0 с огромным количеством примеров на Python).
- Steven Bird, Ewan Klein, and Edward Loper. Natural Language Processing with Python(Книга про обработку текста с помощью библиотеки NLTK для языка Python. Доступна на сайте NLTK)
- Daniel Jurafsky, James H. Martin. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition (Одна из лучших книг про обработку текстов)
- Christopher D. Manning, Hinrich Schütze. Foundations of Statistical Natural Language Processing (Книга содержит хорошие примеры применения машинного обучения для обработки текстов)