ПРАКТИЧЕСКОЕ ЗАДАНИЕ №2. ОСЕНЬ 2021

ПОСТАНОВКА ЗАДАЧИ

Целью работы является разработка метода, позволяющего выделять сущности трёх типов (именованные сущности двух типов и даты) в русскоязычных новостных текстах. Именованной сущностью считается слово или словосочетание, обозначающее предмет или явление, выделяющее этот предмет или явление из ряда однотипных предметов или явлений. Именованная сущность всегда имеет референт (объект, которому принадлежит это имя).

ТИПЫ ИМЕНОВАННЫХ СУЩНОСТЕЙ

- **Дата** (метка **DATE**) – явная текстовая отсылка к некоторому конкретному дню, или промежутку времени большего размера (неделя, месяц, год, ...). Обычно упоминание даты состоит из нескольких опциональных частей, следующих в произвольном порядке: дня недели, числа, месяца и года. Также упоминанием даты являются относительные даты (вчера, завтра и т.п.).
- **Личность** (метка **PERSON**) – имя конкретного человека (реального или вымышленного) или существа, наделенного свойствами человека
- **Организация** (метка **ORGANIZATION**) – именованные группы людей, объединенных для достижения какой-либо цели. Организациями являются в том числе юридические лица, органы государственной/муниципальной власти, политические объединения, спортивные команды, преступные группировки и т.д., и т.п.

РЕШЕНИЕ ЗАДАЧИ

ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ

В рамках решения второго задания практикума предполагается использование методов машинного обучения с учителем. Для обучения метода требуется придумать признаки и дать ему на вход правильные примеры - обучающий корпус.

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.

РАЗМЕТКА ОБУЧАЮЩЕГО КОРПУСА

Для разметки корпуса необходимо зарегистрироваться на сайте https://2021-2.tpc.ispras.ru (или использовать учетные данные первого практического задания). Пожалуйста, вводите правильные данные, так как именно они будут использоваться при выставлении зачетов.

Далее система будет показывать тексты новостных статей. В каждом предложенном тексте необходимо выделить **все** именованные сущности заданных типов.

В систему загружено около 1000 текстов новостей. Длина каждого текста не менее 1000 и не более 3000 символов. Каждому человеку предлагается разметить не менее 40 случайно выбранных текстов. Система перестает предлагать текст для разметки, если он был размечен 3 разными людьми. Информацию о различиях в разметке можно использовать при обучении алгоритмов.

После того, как будут размечены не менее 40 текстов, появится кнопка, позволяющая скачать размеченные тексты, и станет доступна загрузка решений.

Рекомендуется размечать максимально честно, так как от этого будет зависеть качество всех моделей.

Правила разметки именованных сущностей

Общие правила разметки

- В тексте должны выделяться все вхождения именованных сущностей, в том числе аббревиатуры, транслитерированные названия.
- При определении типа сущности обязательно должен учитываться контекст. Например, строка “facebook” является сущностью типа организация, если использована для обозначения компании, а не социальной сети.
- Именованной сущностью считается наиболее длинная цепочка последовательных слов, отражающих имя. В составе именованной сущности не должны содержаться слова, которые не относятся к имени.
- В состав именованной сущности не должны включаться слова-дескрипторы (слова-индикаторы), если они не являются частью названия. Например, слово “компания” для именованных сущностей типа “Организация”, должности для сущностей типа “Персона”.
- Границы именованных сущностей должны совпадать с границами слов. Исключениями могут являться только случаи опечаток (пропущенных пробельных символов)
- Знаки препинания должны быть включены в состав именованной сущности, только если они являются частью имени. При этом название организации, полностью включенное в кавычки, выделяется вместе с кавычками.

Особые правила разметки сущностей типа “Дата”

- В состав дат включаются предлоги, показывающие является дата началом или концом какого-то интервала времени.
- упоминание даты обычно состоит из нескольких опциональных частей, следующих в произвольном порядке: дня недели, числа, месяца и года.
- упоминанием даты являются относительные даты (вчера, завтра и т.п.).
- в упоминание даты включаются предлоги, стоящие непосредственно перед датой
- в упоминание даты включаются все составные части упоминания
- идущие подряд относительные и абсолютные упоминания дат разделяются на два упоминания
- упоминания разных дат, содержащие общие слова, объединяются в одну дату
- упоминанием даты не являются:
  - интервал времени, не привязанный к некоторому конкретному дню
  - Часть названия чего-либо
  - Относительные даты, для определения абсолютного значения которых требуются дополнительные знания о мире

Особые правила разметки сущностей типа “Персона”

- упоминания разных людей, содержащие общие слова размечаются как разные именованные сущности. Например, “Братья Джоэл и Итан Коэны”.
- упоминания групп людей, объединенные фамилией, размечается как сущность. например, “братья Кличко”.
- упоминания персон внутри названий (премий, книг, фильмов, …) размечаются как сущности типа Персона.
- упоминания персон внутри названий организаций не размечаются (при этом размечается название организации).
- сущности типа Персона включают в себя титулы, но не включают должности.

Тренировочный корпус

Тренировочный корпус будет доступен для скачивания в формате json. Для извлечения информации из этого файла рекомендуется использовать стандартную библиотеку Python с одноименным названием. Для синхронизации обучения и тестирования в течение недели, корпус будет состоять из новостей, размеченных автором классификатора, и всех текстов, размеченных в течение предшествующей недели.

Тестирование

Вместе с кнопкой скачивания тренировочного корпуса появится ссылка на форму для загрузки решения и личную страницу со статистикой. На личной странице 2021-2.tpc.ispras.ru/submissions/nerc находится статистика со всеми результатами в т.ч. результатами последнего тестирования (дата, метрики качества).

На странице 2021-2.tpc.ispras.ru/results доступны результаты всех участников. Таблица обновляется раз в неделю.

Загрузка решения

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

- Решение в файле ***solution.py***. В файле должен содержаться класс ***Solution***. В классе должны присутствовать методы:
  - `train(self, train_corpus: List[Tuple[str, Dict[str, Set[Tuple[int, int, str]]]]]) -> None`, где ***train_corpus*** – это список пар <текст; разметка>. Разметка представляет собой отображение идентификатора автора разметки во множество именованных сущностей. Каждая именованная сущность представлена тройкой. Границы сущности в тексте задаются как $[start; end)$.
  - `predict(self, news: List[str]) -> List[Set[Tuple[int, int, str]]]`, который получает на вход список текстов и возвращает список множеств именованных сущностей.
- описание применяемых методов в файле ***description.txt***. Пожалуйста, напишите подробное описание, какие методы и признаки использовались. Это описание будет выложено вместе с решением после завершения курса.
- все используемые ресурсы, необходимые для корректной работы метода. Названия файлов не должны совпадать с названиями embedding моделей.

Результаты тестирования появятся на личной странице, как только закончится обучение и тестирование. В течение недели студенты не видят прогресс своих коллег и могут посмотреть только свой результат.

ПРИМЕР РЕШЕНИЯ, ВОЗВРАЩАЮЩЕГО ПУСТОЙ РЕЗУЛЬТАТ

```
class Solution:
    def train(self, train_corpus):
        pass

    def predict(self, news):
        return [set() for _ in news]
```

Практические аспекты

Решения должны быть написаны на языке Python (версия 3.8.10). Можно использовать все стандартные библиотеки, а также:

- nltk==3.6.5 - инструменты для обработки текстов
- scikit-learn=1.0 - алгоритмы машинного обучения
- numpy==1.21.2 - работа с многомерными массивами
- tensorflow==2.6.0, pytorch==1.9.1, keras==2.6.0 – библиотеки для работы с искусственными нейронными сетями
- transformers==4.11.3
- pymystem3==0.2.0
- pandas==1.3.4
- regex==2021.8.3
- gensim==4.1.2

В случае необходимости использования дополнительных библиотек, сообщите об этом организаторам практикума (библиотеки будут добавлены для всех студентов).

Дополнительно всем решениям доступны предобученные embedding модели (сами модели можно найти на странице практикума). В случае необходимости использования дополнительных моделей, сообщите об этом организаторам практикума (модели будут добавлены для всех студентов). Embedding модели доступны из рабочей директории и лежат в папке /embeddings ('/embeddings/<название файла>').

Доступ в Интернет на проверяющей машине закрыт.

Предобученные эмбеддинги

word2vec эмбеддинги (gensim, txt format):

- w2v_lower_size50_window5.txt
- w2v_lower_size50_window10.txt
- w2v_lower_size100_window5.txt
- w2v_lower_size100_window10.txt
- w2v_size50_window5.txt
- w2v_size50_window10.txt
- w2v_size100_window5.txt
- w2v_size100_window10.txt

BERT эмбеддинги (deep pavlov)

- rubert_cased_L-12_H-768_A-12_pt.tar.gz
- rubert_cased_L-12_H-768_A-12_v2.tar.gz

RuBERT (СберAI)

- ruBert-base/ruBert-base.bin, ruBert-base/config.json, ruBert-base/vocab.txt

ОГРАНИЧЕНИЯ

- Каждую неделю можно послать не более 5 решений.

Внимание! Итоговое тестирование будет проводиться на последнем загруженном решении.

- Размер загружаемого архива не должен превышать 15Мб.
- Время тестирования одного решения (обучение + предсказание) не должно превышать 30 минут.
- На проверяющей машине доступно 16 Гб оперативной памяти.

В связи с первым ограничением, для тестирования на локальной машине рекомендуется использовать метод перекрестной проверки.

ОЦЕНКА КАЧЕСТВА

Для оценки качества используется усреднение micro-averaged F1-мер для каждого типа именованных сущностей. micro-averaged F1-мера вычисляется как среднее гармоническое micro-averaged precision и micro-averaged recall.

$$Precision^t = \frac{|predicted^t \cap expected^t|}{|predicted^t|};$$

$$Recall^t = \frac{|predicted^t \cap expected^t|}{|expected^t|};$$

$$F_1^t = 2\frac{Precision^t \cdot Recall^t}{Precision^t + Recall^t};$$

$$F_1 = \frac{1}{|T|}\sum_{t \in T} F_1^t;$$

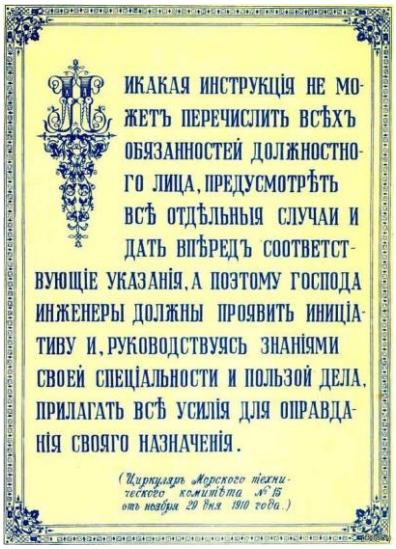
где T – множество типов именованных сущностей; $predicted^t$, $expected^t$ – множества предсказанных и ожидаемых сущностей типа t.

BASELINE РЕШЕНИЕ

В качестве первого baseline решения предлагается модель, основанная на словарях. Во всех текстах числа заменяются нулями. Во время обучения формируются словари, отображающие сущность в количество вхождений. На этапе предсказания тексты токенизируются и осуществляется поиск наиболее длинных вхождения сущностей, встречавшихся более двух раз в сформированных словарях.

В качестве второго baseline решения предлагается модель, основанная на искусственных нейронных сетях. Задача представляется как задача разметки последовательности слов (используется BIO кодировка). Каждый текст делится на предложения и слова (Для токенизации используются регулярные выражения). В качестве признаков используются доступные embedding’и слов текста. Закодированные предложения подаются в BiLSTM-CRF сеть.

КОММЕНТАРИИ ОТНОСИТЕЛЬНО ПРАВИЛ РАМЗЕТКИ



1. Формализация задания

Вопросы относительно правил разметки сложных случаев могут остаться без ответа (в этом случае необходимо руководствоваться жизненным опытом, внутренней интуицией и здравым смыслом).

Все "определения", данные в рамках описания (попытки формализации) правил разметки не являются точными и полностью корректными. Необходимо всегда проецировать их на здравый смысл. Даже если посмотреть на рекомендации аннотирования распространенных бенчмарков (если вообще удастся их [рекомендации] обнаружить) для оценки качества задачи NERC, то можно заметить, что типы размечаемых сущностей задаются неформально, иногда просто группой примеров. Человек, выполняющий разметку обучающего (тестового) корпуса, самостоятельно интерпретирует типы сущностей. Для определения степени согласия разметки важно, чтобы разметка выполнялась индивидуально. Разрешать спорные ситуации коллективно заранее необходимо лишь для очень частых (распространенных) спорных случаев и технических особенностей разметки (включение пунктуации в сущности, варианты разметки для склеенных или пересекающихся сущностей и т.п. [Большинство таких особенностей уже учтены в правилах разметки сущностей отдельных типов]). Система разметки настроена так, что одинаковые тексты предлагаются для разметки студентам несколько раз (каждый текст может быть размечен вплоть до 3 раз). При обучении модели вам будут доступны все варианты разметки. Кроме того формат представления обучающих данных предусматривает возможность идентификации разметки различных текстов одним автором (по идентификатору в системе разметки). Вы можете пробовать различные способы борьбы с коллизиями разметки (это часть процесса обучения модели, которая может существенным образом влиять на итоговое качество модели). При разметке тестового корпуса асессорам было доступно подмножество правил разметки, представленных в задании (в результате анализа коллизий набор правил был расширен) и степень согласия даже при "плохо поставленной задаче" оказалась довольно высокой. Каждый из 99 тестовых документов был размечен 3 асессорами. Более 85% сущностей были размечены одинаково всеми тремя асессорами (эта оценка скорее всего увеличится для полного набора правил, представленных в формулировке задания) (многие ошибки были чисто техническими + невнимательность. Действительно сложных случаев было менее 1%). В связи с этим с этого момента (для стимуляции самостоятельности разметки) вопросы относительно правил разметки сложных случаев могут остаться без ответа (в этом случае необходимо руководствоваться жизненным опытом, внутренней интуицией и здравым смыслом).

2. Методология оценки качества.

Все модели (включая baseline) тестируются в одинаковых условиях.

2.1 Разделение train|test выборки

В случае, если бы задачей второго задания практикума стояло обучение какого-либо метода машинного обучения, то требование совпадения распределений тренировочной и тестовой выборок можно было бы обосновать хоть каким-то образом (хотя бы потому, что машинное обучение должно работать в этих случаях и не должно [хотя может] в других). Однако задачей второго задания практикума является решение проблемы извлечения информации из текстов (а именно NERC). В такой постановке задачи никаких ограничений на распределения в тренировочной и тестовой выборках нет (хотя мы старались сделать так, чтобы распределения были похожи: в тестовой части тоже новости, тоже на русском языке, даже из +- того же

временного интервала, ожидаются сущности тех же типов [размеченные по тем же правилам]). Именно в такой постановке задачи на наш взгляд оценка методов решения задачи приближена к реальным условиям: постановка задачи "на словах" (при этом возможны ответы на некоторые вопросы), апробация полученных результатов на новых данных (в данном случае мы ограничиваемся просто закрытым test, хотя можно также кроме закрытого test использовать дополнительную закрытую часть корпуса для "финальной" оценки моделей [этот способ оценки не используем, поскольку для студентов результаты могут оказаться слабо предсказуемыми и больше похожими на случайные]), оценка полученных результатов другими людьми (именно поэтому train и test размечается независимо разными группами асессоров).

2.2 ПОПОЛНЕНИЕ ОБУЧАЮЩЕГО КОРПУСА

Процесс разметки обучающего корпуса растянут во времени (к сожалению это данность). Разные студенты в разное время приступают к выполнению задания и выполняют его с разной скоростью. Поскольку в одиночку разметить корпус, достаточный для обучения модели (в данном случае мы не рассматриваем постановки задачи few- и one-shot learning), очень тяжело, в качестве обучающего корпуса используется разметка всех студентов (объединение происходит каждую неделю строго перед запуском методов для еженедельной оценки). Для сохранения справедливости студенту всегда доступна вся разметка, которую он выполнил самостоятельно. Именно из этих условий родилось не совсем очевидное правило формирования обучающего корпуса: каждому студенту доступна собственная разметка + разметка остальных студентов за предыдущую неделю.

Поскольку каждую неделю обучающий корпус меняется, итоговое качество моделей (даже если метод не менялся) может изменяться (в том числе результат baseline). Однако поскольку в недельных тестах (и при подведении итогов) обучающий корпус у всех методов одинаковый, качественную оценку методов можно считать справедливой.