|  |  |
| --- | --- |
| Урок | Домашнее задание |
| 1 | Формат именования файла домашней работы: FIO\_NLP\_HW\_N.ipynb, где N - номер домашнего задания  \*\*  Осуществим предобработку данных с Твиттера, чтобы отчищенный данные в дальнейшем использовать для задачи классификации. Данный датасет содержит негативные (label = 1) и нейтральные (label = 0) высказывания.  Для работы объединим train\_df и test\_df.  Задания:  1) Удалим @user из всех твитов с помощью паттерна "@[\w]\*". Для этого создадим функцию:  - для того, чтобы найти все вхождения паттерна в тексте, необходимо использовать re.findall(pattern, input\_txt)  - для для замены @user на пробел, необходимо использовать re.sub()  2) Изменим регистр твитов на нижний с помощью .lower().  3) Заменим сокращения с апострофами (пример: ain't, can't) на пробел, используя apostrophe\_dict. Для этого необходимо сделать функцию: для каждого слова в тексте проверить (for word in text.split()), если слово есть в словаре apostrophe\_dict в качестве ключа (сокращенного слова), то заменить ключ на значение (полную версию слова).  4) Заменим сокращения на их полные формы, используя short\_word\_dict. Для этого воспользуемся функцией, используемой в предыдущем пункте.  5) Заменим эмотиконы (пример: ":)" = "happy") на пробелы, используя emoticon\_dict. Для этого воспользуемся функцией, используемой в предыдущем пункте.  6) Заменим пунктуацию на пробелы, используя re.sub() и паттерн r'[^\w\s]'.  7) Заменим спец. символы на пробелы, используя re.sub() и паттерн r'[^a-zA-Z0-9]'.  8) Заменим числа на пробелы, используя re.sub() и паттерн r'[^a-zA-Z]'.  9) Удалим из текста слова длиной в 1 символ, используя ' '.join([w for w in x.split() if len(w)>1]).  10) Поделим твиты на токены с помощью nltk.tokenize.word\_tokenize, создав новый столбец 'tweet\_token'.  11) Удалим стоп-слова из токенов, используя nltk.corpus.stopwords. Создадим столбец 'tweet\_token\_filtered' без стоп-слов.  12) Применим стемминг к токенам с помощью nltk.stem.PorterStemmer. Создадим столбец 'tweet\_stemmed' после применения стемминга.  13) Применим лемматизацию к токенам с помощью nltk.stem.wordnet.WordNetLemmatizer. Создадим столбец 'tweet\_lemmatized' после применения лемматизации.  14) Сохраним результат предобработки в pickle-файл. |
| 2 | Приложил к 1 лекции все методички по курсу  \*\*  Домашнее задание 2 варианта на выбор:  1 вариант (более легкий ) по ссылке https://gbcdn.mrgcdn.ru/uploads/asset/2800508/attachment/447973cb1f55a137a8a7eb4b175f4b47.pdf  2 вариант (рекомендую его, он более объемный, но и больше навыков отрабатывается) по описанию ниже:  все материалы для выполения дз в sem2.ipynb он находится в архиве Lesson-2.zip  для скачивания данных пользуйтесь этими ссылками  https://disk.yandex.ru/i/v5HM-ENiGXZVpQ  https://disk.yandex.ru/i/koR5eMCToCZS2Q  Задание 1.  Задание: обучите три классификатора:  1) на токенах с высокой частотой. ([ :N] частотных слов)  2) на токенах со средней частотой ( [N:N+X] частотных слов)  3) на токенах с низкой частотой( [N+X :] частотных слов)  Сравните полученные результаты, оцените какие токены наиболее важные для классификации.  Задание 2.  найти фичи с наибольшей значимостью, и вывести их  Задание 3.  1) сравнить count/tf-idf/hashing векторайзеры/полносвязанную сетку (построить classification\_report)  2) подобрать оптимальный размер для hashing векторайзера  3) убедиться что для сетки нет переобучения |
| 3 | Скачиваем датасет  !wget https://github.com/ods-ai-ml4sg/proj\_news\_viz/releases/download/data/gazeta.csv.gz  Или данные можно скачать тут https://github.com/IlyaGusev/gazeta  Пример работы с ним:  from corus import load\_ods\_gazeta path = 'gazeta.csv.gz' records = load\_ods\_gazeta(path) next(records)  что надо сделать:  на основе word2vec/fasttext реализовать метод поиска ближайших статей (на вход метода должен приходить запрос (какой-то вопрос) и количество вариантов вывода к примеру 5-ть, ваш метод должен возвращать 5-ть ближайших статей к этому запросу)  Проверить насколько хорошо работают подходы |
| 4 | Задача тематическое моделирование  продолжаем исследование датасета с твитами  Скачиваем датасет (источник): положительные, отрицательные.  или можно через ноутбук  !wget https://www.dropbox.com/s/fnpq3z4bcnoktiv/positive.csv  !wget https://www.dropbox.com/s/r6u59ljhhjdg6j0/negative.csv  как альтернатива можно скачать данные из Роспотребнадзора  https://zpp.rospotrebnadzor.ru/Forum/Appeals  для этого берём ноутбук parse\_rospotrebnadzor.ipynb  устанавливаем количество скачанных страниц больше не 50-сят хотябы 500 и для анализа берём только вопросы так как ответы есть не всегда  что надо сделать  1. объединить в одну выборку (это только для твитов), для роспотребнадзора сформировать датасет из вопросов  2. провести исследование и выявить тематики о которых говорят в твитах (для твитов), а для роспотребнадзора так же выявить тематики о которых люди пишут проанализировать  3. сделать визуализацию кластеров тематик  4. проинтерпритировать получившиеся тематики |
| 5 | Задание по ссылке  https://docs.google.com/document/d/1YxTvITBz9tQANE9lO6T4FJr2sm9GJUZo/edit# Тема «POS-tagger и NER» **Задание 1. Написать теггер на данных с русским языком**   1. проверить UnigramTagger, BigramTagger, TrigramTagger и их комбинации 2. написать свой теггер как на занятии, попробовать разные векторайзеры, добавить знание не только букв но и слов 3. сравнить все реализованные методы, сделать выводы   **Задание 2. Проверить, насколько хорошо работает NER**  **Данные брать из http://www.labinform.ru/pub/named\_entities/**   1. проверить NER из nltk/spacy/deeppavlov. 2. написать свой NER, попробовать разные подходы.    1. передаём в сетку токен и его соседей.    2. передаём в сетку только токен.    3. свой вариант. 3. сравнить свои реализованные подходы на качество — вывести precision/recall/f1\_score. |
| 6 | Взять ноутбук colab\_text\_classification\_part1.ipynb который разбирали на занятии и добавить пункты которые мы пропустили  1. Проверьте повысилось ли качество на стандартных подходах при лемматизации/и без неё  2. Удалите/(замените на тег) из текстов сущности(имена, локации, что-то ещё). Запустите классификатор и модельки на сеточках  3. Сделайте выводы |
| 7 | Берем отзывы за лето (из архива с материалами или предыдущего занятия)  Учим conv сеть для классификации  Рассмотреть 2-а варианта сеточек:  Инициализировать слой tf.keras.layers.Embedding по умолчанию (ну то есть вам ничего не делать с весами)  Инициализировать tf.keras.layers.Embedding предобученными векторами взять к примеру с https://rusvectores.org/ru/  Сравнить две архитектуры с предобученными весами и когда tf.keras.layers.Embedding обучается сразу со всей сеточкой, что получилось лучше |
| 8 | Практическое задание к уроку 8. Рекуррентные нейронные сети RNN LSTM GRU  Данные берем отызывы за лето  На вебинаре мы говорили, что долгое время CNN и RNN архитектуры были конурируещими выяснить какая архитектура больше подходит для нашей задачи  построить свёрточные архитектуры  построить различные архитектуры с RNN  построить совместные архитектуры CNN -> RNN или (RNN -> CNN) |
| 9 | Урок 9. Языковое моделирование  Задание  Разобраться с моделью генерации текста, собрать самим или взять датасет с вебинара и обучить генератор текстов |
| 10 | Урок 10. Машинный перевод. Модель seq2seq и механизм внимания  Разобраться с моделью перевода (без механизма внимания) как она устроена, запустить для перевода с русского на английский (при желании можно взять другие пары языков)  В данном практическом задании напишу и обучу модель перевода с русского на английский  на библиотеке pytorch. Добавлю в неё механизм внимания, так как это сложнее и интереснее,  а также рекуррентную сеть энкодера сделаю двунаправленной. |
| 11 | Разобраться с моделью перевода (с механизмом внимания) как она устроена, запустить для перевода с русского на английский (при желании можно взять другие пары языков) |
| 12 | Задание  Реализовать суммаризацию текста  Взять тот же датасет, который был на вебинаре и предобученную модель для задачи суммаризации  Проверить насколько хорошо она суммаризирует  2.(дополнительно) Сделать генерацию заголовков для статьи (обучить модель для генерации заголовков) |
| 13 | Задание 13  Взять датасет https://huggingface.co/datasets/merionum/ru\_paraphraser решить задачу парафраза  (дополнительно необязательная задача)на выбор взять https://huggingface.co/datasets/sberquad https://huggingface.co/datasets/blinoff/medical\_qa\_ru\_data натренировать любую модель для вопросно ответной системы как альтернатива можно взять любой NER датасет из https://github.com/natasha/corus#reference и обучить NER |
| 14 | Задание 14  Задание взять данные из https://www.kaggle.com/datasets/mrapplexz/bashim-quotes обучить модель GPT для генерации своих цитат  взять новостные данные из https://github.com/natasha/corus load\_lenta2 нам понадобиться сам текст и заголовок обучить модель T5/ или GPT для генерации заголовков для статей |
| 15 | Текстовое задание:  Задание  Продумать тему курсового проекта.  Подготовить данные для обучения.  Продумать интенты на которые вы будете обучаться.  Для сдачи задания необходимо написать:  На какую тему ваш бот будет общаться (на каких датасетах планируете обучать модели)  Написать на какие интенты вы будете обучать бота  Примеры интентов  болталка  погода в городе  время в городе  праздники  фильмы  покупка товара  можно добавить что-то своё  Из обязательных (всего обязательно 3-ри интента):  ваш бот дожен работать в режиме  болталка  на ваш выбор |
| 16 | Создать чат-бота.  Прислать код  Основное задание - сделать проактивный алгоритм, который на фразу дает релевантный ответ.  Реализация в телеграмм опционально |
|  |  |
|  |  |