Лабораторная 5 - текст

[github classroom task](https://classroom.github.com/a/OoPO6PN7)

Порядок сдачи и защиты лабы описан [тут](https://docs.google.com/document/d/1pj0LIO8kuOzqxgxW-dfZkQFhwZ9dkQ1hIxg5Va39x7U/edit?usp=drive_link)

Дедлайн указан в таблице с баллами

# Тайный метод быстрой сдачи лабы на полный балл

1. Проверьте, что код загружен в main ветку
2. Проверьте, что ничего из раздела Популярные замечания не нужно вашему коду
3. Перед вопросом в чат проверить faq лабы
4. Если видите в чате вопрос 2+ раз, пингуйте практика добавить вопрос в faq

# Задание

## Данные

**Кратко**: соберите корпус текстов

**Подробно**:

Использование собранных самостоятельно данных даёт 5 бонусных (то есть реально 5, то есть почти 50% доп баллов за лабу) баллов

Можно использовать датасет из 1 лабы. Можете спарсить доп данные с сайта. Либо можете выбрать другой сайт и собрать интересные текстовые данные (даже через api)

Интересные данные - это что-то нишевое и близкое вам. Например, редкими и интересными в 1 лабе были датасеты по матчам в cs, гонки f1, кроссовки, авиакатастрофы, игры на консоли, продукты из перекрестка, данные о соревах по атлетике. Менее интересные данные - парсинг крупных сайтов без фильтров: все товары с авито, все квартиры с циана

Хотите использовать аморальные/заблокированные сайты - используйте. Знаете сайт с интересными текстами, но их нужно будет разметить на классы? Это не проблема, это автоматизируется

Можно использовать датасет из интернета, но тогда бонусных баллов (и респекта) и не будет

### Идеи датасетов

**Кратко**: можно придумать любую шизу и разметить датасет

**Подробно**:

Использовать конкретно эти - уже не так интересно. Но для вдохновения подойдет

* комментарии/посты в соц сети  
  Внутри каждой соцсети бесконечно много сообществ, внутри каждого есть дихотомия, которую можно научиться выделять автоматически. Например склонность к какой-то позиции: плоскости земли/правоты общественного движения/количества гендеров/etc. Или наличие свойства в тексте: токсичность/эмоциональность (от негативного до позитивного)/etc. Можно выделять какие слова склонили детектор в сторону токсичности например. Или соответствие текста нормам конфуцианской морали
* распознанный текст видео/песен  
  то же, что про комменты. Ютуб генерирует субтитры автоматически, тексты песен давно разобраны
* отзывы о чем угодно  
  можно кластеризовать отзывы об инфоцыганских курсах и понять что волнует людей больше всего. Можно создать статистику про отзывы из delivery club про любимый ресторан. Можете исследовать отзывы о любимой игре в steam или сообщения из чатов во время матча в dota 2
* твиты  
  можно исследовать как изменилось общественное мнение о любом вопросе: скачать твиты по ключевому слову и изучить sentiment/кластеризацию
* описания   
  можно по тексту об одежде предсказать ее рейтинг или по описанию или количество лайков на посте в паблике, где несколько тысяч постов

## Задача

Можете взять любую NLP задачу:

* Просто
  + классификация
  + регрессия
  + выделение эмоций (sentiment/toxic detection/etc)
* Средне:
  + моделирование языка
  + генерация
  + machine translation
  + topic modelling
* Сложно:
  + ответы на вопросы
  + диалоговые модели
  + суммаризация

Балл за лабу не зависит от задачи

## План действий

**Кратко**: разметьте датасет с помощью LLM и обучите NLP модель, измерьте метрики  
**Подробно**:

1. Получите доступ к api любой LLM  
   Если у вас нет аккаунта open ai, то время его завести или найти качественный сервис-посредник. Программный доступ к LLM полезен во многих ситуациях. Способ для богатых - захостите LLM локально. Выведите в ноутбуке ответ на любимый вопрос из теорминов в стиле, который вам нравится (хоть в стихотворном)
2. Соберите датасет из текстов  
   Берите имеющийся или парсите новый. Вы уже разбираетесь в парсинге, поэтому это теперь не сложная задача. Можно использовать API
3. (если нет разметки) Разметьте данные  
   Используйте LLM из пункта 1. Напишите [подробный промпт](https://www.promptingguide.ai/ru/introduction/basics) с описанием задачи. Разметьте весь датасет или такую его часть, которой достаточно для того, чтобы хорошо предсказывать целевую переменную
4. Реализуйте текстовую модель  
   Это может быть простая DL модель - nn.Embedding + nn.LSTM + nn.Linear (разобрано на лекции), сложная предобученная модель, в которой вы честно разобрались или скучная модель из классического обучения (кроме tf-idf и naive bayes, которые уже разобрали на лекции). Использовать реализацию Word2Vec из gensim как свою модель нельзя
5. Обучите модель на размеченном датасете
6. Измерьте качество модели  
   а) Посчитайте метрики на train и test  
   б) (опционально, но интересно) создайте эмбеддинги текстов: используйте выход предпоследнего Linear слоя. Следующий слой модели предсказывает по этому эмбеддингу ответ, поэтому в эмбеддинге на этом слое собрана вся полезная информация. Используйте метод снижения размерности, чтобы отобразить текст на 2д плоскости. Цветом укажите целевую переменную  
   в) (опционально, но полезно) используйте tensorboard для логирования метрик во время обучения модели

Пункты 2, 5, 6 должны занять мало времени, тк понимание материала и большая часть кода у вас есть в 1 и 3 лабах

# Рекомендации

* **не стесняйтесь задавать вопросы в чат курса**
* выбирайте нейросетевые модели, ведь теперь вы наверняка их понимаете
* рекомендуемый размер датасета - больше 1000 текстов. Если будет меньше, то алгоритмы могут плохо обучаться

Для реализации рекомендуются эти python библиотеки:

* pytorch
* tensorflow
* youtokentome для byte-pair encoding

Но вы можете брать и другие

# Популярные замечания

* используйте конфиг файл для констант
* не коммитить .idea
* функции длиннее 20 строк - плохо. Их можно декомпозировать
* в начале кода лабы нет комментария со ссылкой на ваш репозиторий с 1 лабой или нет файлов парсера
* не показано, что модель не переобучена. Сравните метрики на train и test
* в модели нет dropout/batchnorm слоев. Она быстро переобучается из-за этого
* слишком большой размер словаря (>10k) из-за чего модель крайне медленно учится. Лечится c byte-pair encoding
* В архитектуре Embedding + Lstm + Linear у последней части всего один линейный слой. Этого мало для качественного извлечения признаков из результатов обработки текста. Добавьте несколько слоев, перемежая с активациями и dropout, как показано на лекции про feed-forward сети

# Теормин

В каждом вопросе про метрики, методы и тд нужно знать как это рассчитывается и какой геометрический смысл за этим стоит. Большие и страшные формулы зубрить не нужно. Знать формулы гейтов lstm и аналогичные сложные многострочные формулы тоже.

В вопросах 7-14 достаточно знать материал с лекций

В вопросах про модели подразумевается: Как устроена, как обучается, какие требования к данным (к чему уязвима)

1. Что такое эмбеддинг? Является ли строка датасета эмбеддингом? является ли выход линейного слоя нейросети эмбеддингом? Что такое запутанное представление?
2. Проблемы кодирования слов: one-hot, stop-слова, формы слов, **n-gramms**
3. Модель bag of words. Чем cbow отличается от skip-gram
4. Модель tf-idf. Какого размера вектор возвращает, какие данные хранит, как обучается, в чем минусы по сравнению с RNN
5. Что такое корпус, что такое словарь, как размер словаря влияет на алгоритмы. Как уменьшить размер словаря, на что это повлияет
6. Что такое [byte pair encoding](https://huggingface.co/learn/nlp-course/chapter6/5?fw=pt) и как он сокращает размер словаря до минимума, но все равно способен кодировать даже те слова, которые не видел при сборке словаря
7. Что такое рекуррентная ячейка (RNN), почему обрабатывает последовательности любой длины, сколько рекуррентных ячеек в рекуррентной сети, какие векторы принимает и возвращает
8. LSTM ячейка: чем отличается от простейшей RNN, за что отвечают гейты, как борется со взрывом/затуханием градиентов
9. Как с помощью RNN генерировать текст слово за словом
10. Модель (на самом деле тип моделей) seq2seq. Какие задачи может решать, из чего состоят encoder и decoder
11. Attention. Зачем нужен, геометрический смысл
12. Attention. Как получается вес(=важность) эмбеддинга слова по отношению к другому слову?
13. Attention. Меняется ли размер эмбеддинга слова после применения attention-слоя? Меняется ли сам вектор? Как?
14. Как устроена модель GPT-3
15. Какие сейчас существуют LLM? Чем LLM отличается от простых языковых моделей? Сколько весов(=параметров) у GPT-3.5 и у GPT-4? Что означает размер контекста у LLM?
16. Beam search. Зачем нужен, как работает
17. **Метрики BLEU (Bilingual Evaluation Understudy), Perplexity, WER (Word Error Rate). Зачем нужны и как считаются**

# FAQ

1. Мне незачем использовать LLM, можно без нее?  
   Можно, но скучно. LLM - ваш бесплатный разметчик данных. Хорошим промптом вы можете задать ей все необходимые вводные данные для отличной разметки на **любую** тему.   
   Например если хотите предсказать, насколько текст песни соответствует нормам конфуцианской морали
2. Долго получать доступ до openai.com   
   Возьмите любой другой сервис. Сервисов-посредников между юзером и openai сейчас много. Они работают без vpn
3. Что из торча использовать  
   torch.nn.Embeddings для кодирования слов, torch.nn.LSTM для обработки последовательности эмбеддингов, torch.nn.Linear для предсказания слова/класса/etc (зависит от задачи). Не забудьте про слои активации, batchnorm, dropout
4. Как предобрабатывать текст  
   Зависит от токенизатора и задачи. Рекомендуется использовать bpe-токенизацию из youtokentome. При ней токены (=строки) в разном регистре будут считаться разными, поэтому стоит привести к lowercase. Знаки препинания эта библиотека отделит даже если они вплотную к слову. Если токенизация - это деление по пробелу, то токен с восклицательным знаком (без пробела) выделяется как отдельное слово, из-за чего словарь разрастается напрасно