Предобученные языковые модели в задачах обработки текстов

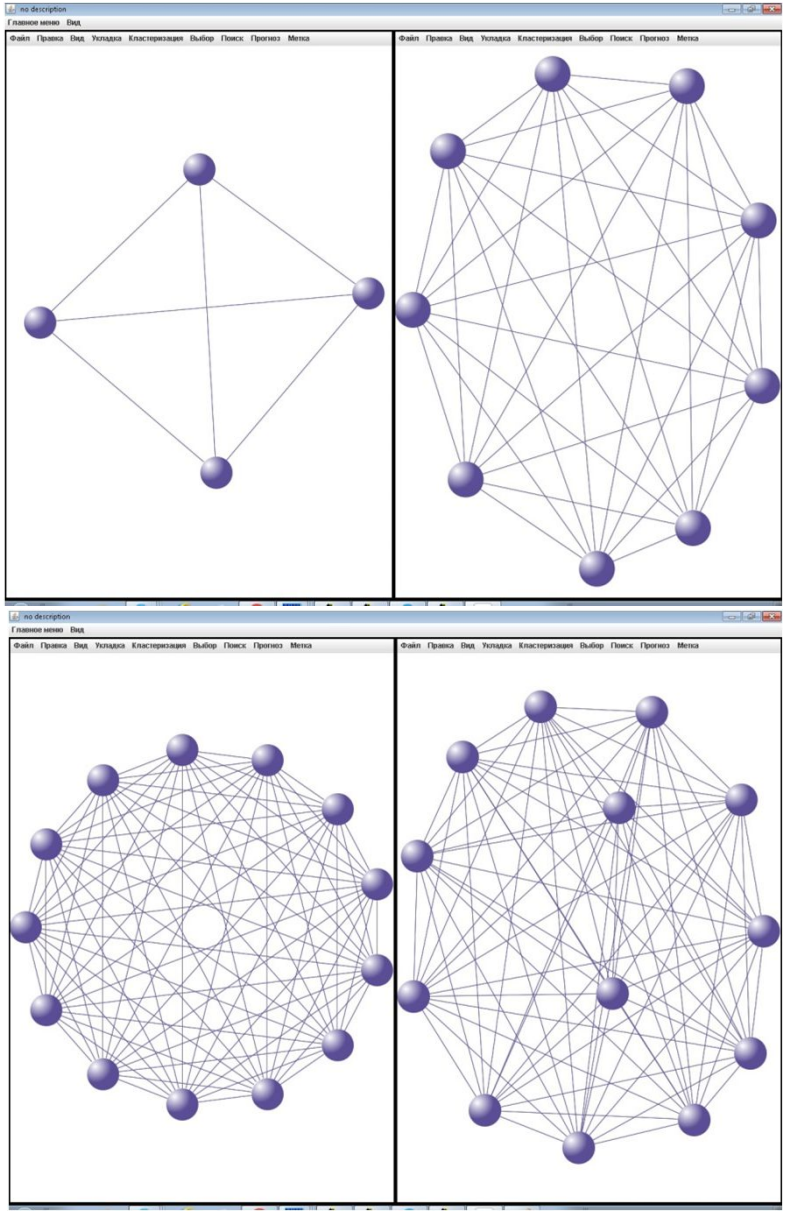
(опыт использования DGX-2
для задач обработки текстов)

Тихомиров Михаил Михайлович, стажер-исследователь НИВЦ МГУ,
аспирант ВМиК МГУ, tikhomirov.mm@gmail.com
Добров Борис Викторович, завлаб НИВЦ МГУ, к.ф.-м.н., dobrov_bv@mail.ru
Лукашевич Наталья Валентиновна, внс НИВЦ МГУ, д.т.н., louk_nat@mail.ru

План

- Общие сведения о нейросетевом подходе решения задач
- Подходы к решению задач обработки текстов
- Предобученные языковые модели
- Опыт использования DGX-2
- Полученные результаты
- Применение полученных результатов

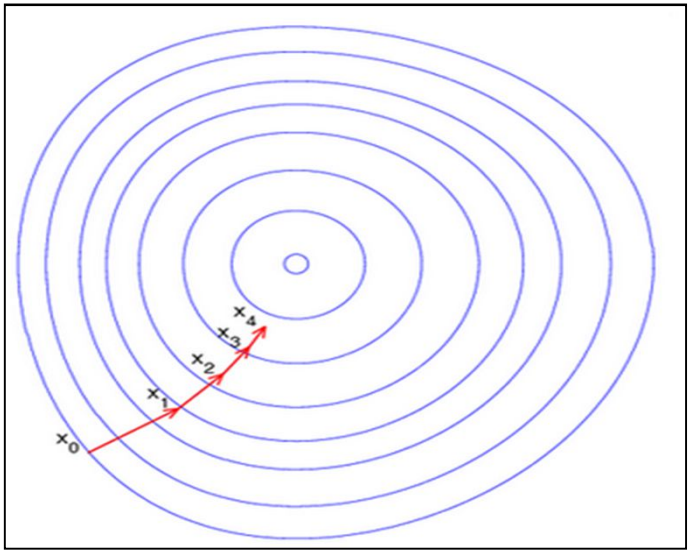
Постановка задачи машинного обучения – как решение задачи оптимизации



$$F(\vec{x}) : \mathbb{X} \to \mathbb{R}.$$

$$F(\vec{x}) \to \min_{\vec{x} \in \mathbb{X}}$$

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F\left(\vec{x}^{[j]}\right)$$



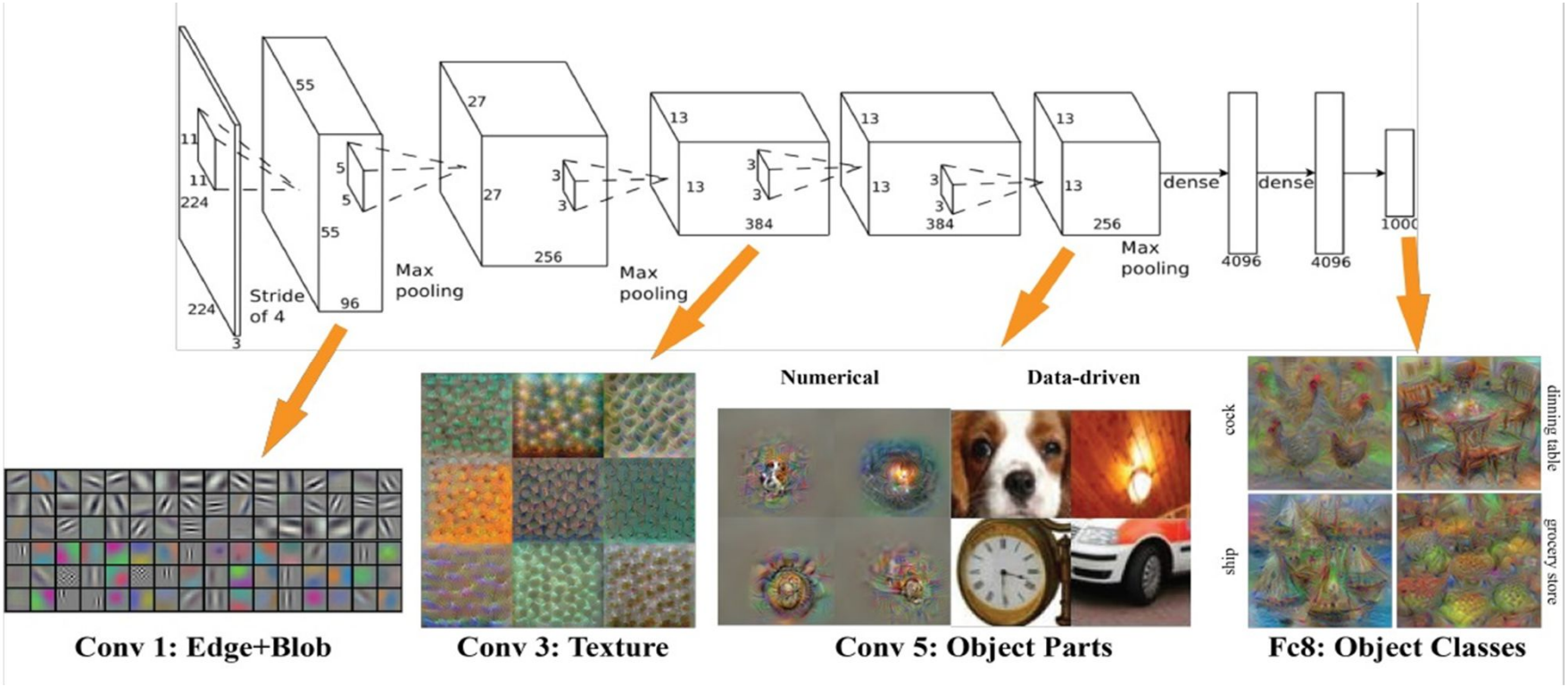
Метод группового учёта аргументов (МГУА) vs. глубокое обучение

А.Г.Ивахненко – еще в 1971 году использовался для обучения восьмислойной нейронной сети – самая ранняя реализация глубокого обучения

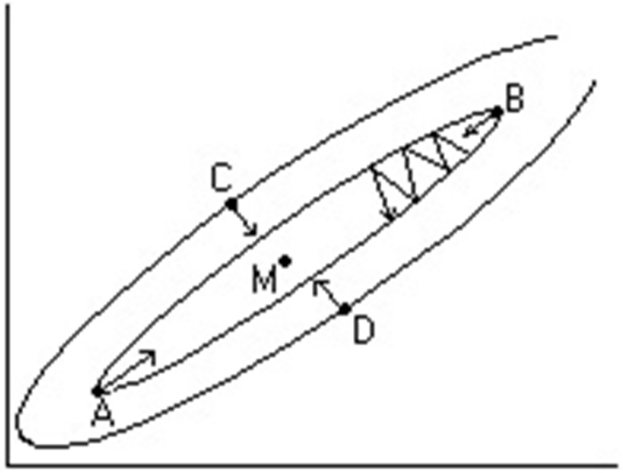
$$Y(x_1,\ldots,x_n) = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=i}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} a_{ijk} x_i x_j x_k + \cdots$$

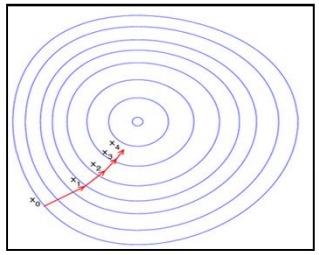
$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_n x^n, \quad a_i \in \mathbb{R}$$

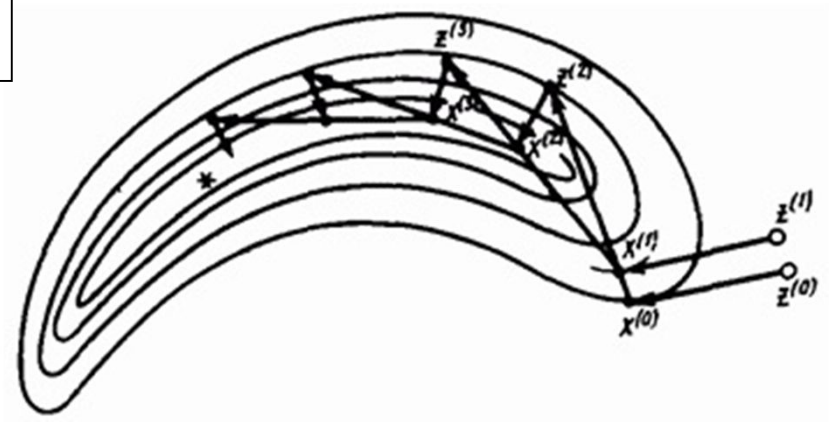
$$P(x) = a_0 + x(a_1 + x(a_2 + \cdots x(a_{n-1} + a_n x)\ldots))$$



Стохастический градиент, «батчи»







«Проклятие» оврагов
➔ идти не по градиенту,
а по возможным направлениям

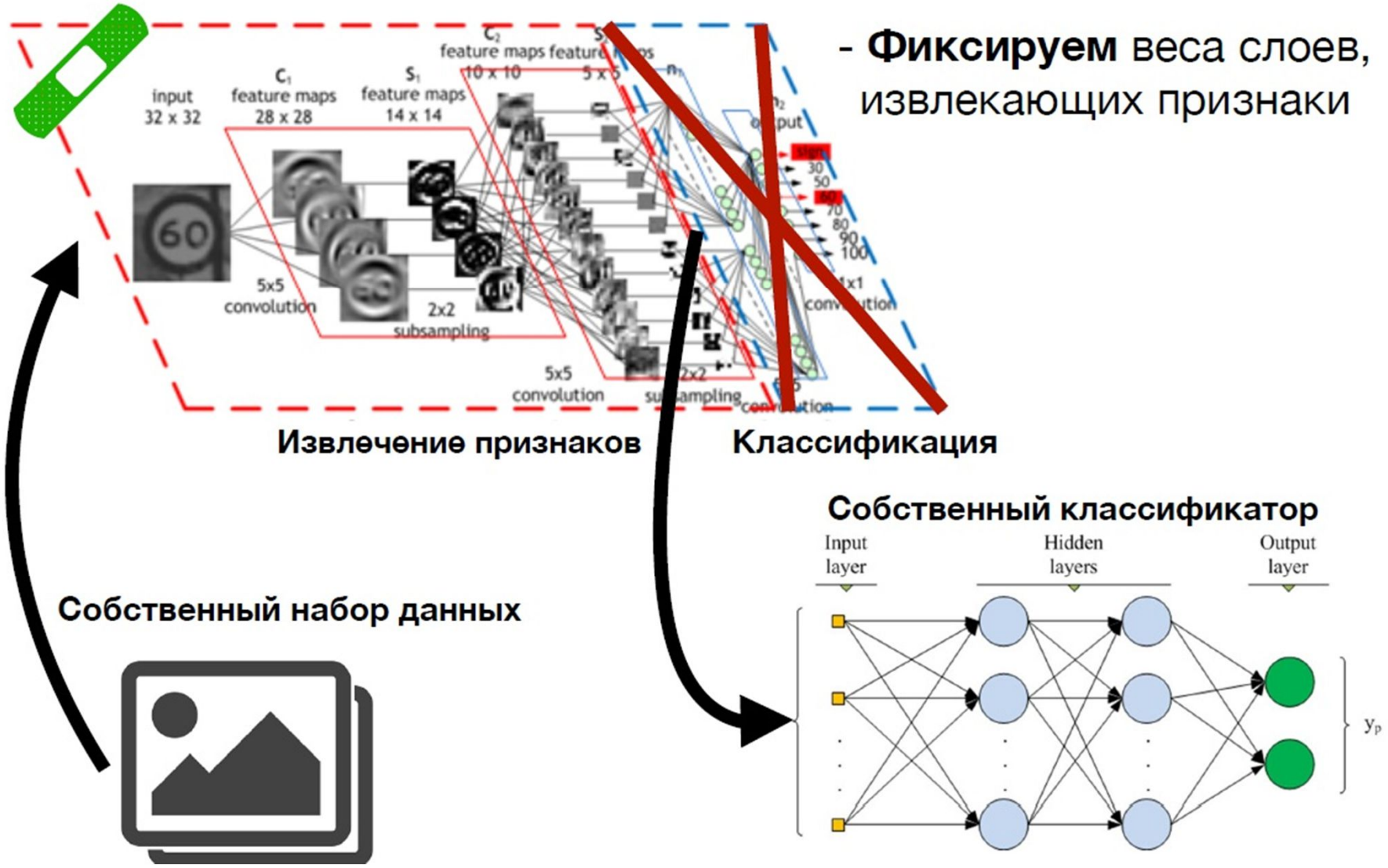
Учет истории (модель локального поведения функции) – «метод тяжелого шарика» – «тензорные поезда»

Оптимизируемая функция – сумма большого количества невязок по обучающим примерам

$$\sum_{m=1}^{M}\sum_{n=1}^{N^L}(d_n^m - z_n^{L,m})^2$$

Если среди примеров много схожих – можно случайно выбирать примеры и делать шаг оптимизации не вычисляя полностью функцию

Перенос обучения (Transfer learning)



Современная инфраструктура Data Science

- Публикации
  - журналы
  - конференции
  - Arxiv.org (препринты) !!!
- Вычислительные архитектуры на GPU
- Методология переноса обучения
- Свободно доступные библиотеки машинного обучения
- Свободно доступный код различных проектов: Github
- Свободно доступные наборы данных для обучения
- Широкое распространение Python
- Потенциально доступные вычислительные мощности

В результате можно требовать от студента принести через неделю реализацию для таких задач, которые ранее решались десятилетиями целыми институтами
(синтаксический анализ, распознавание речи, фрагментация видео, распознавание изображений, …)

Языковые модели

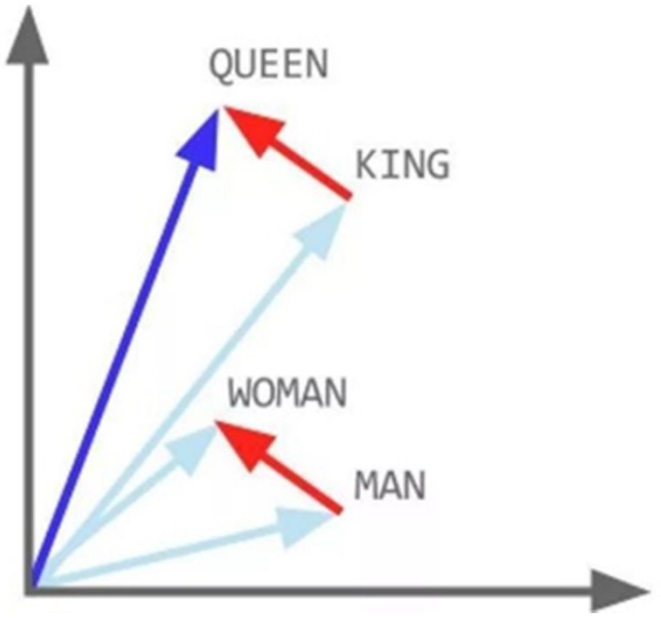
И вот уже трещат морозы
И серебрятся средь полей…
Читатель ждет уж рифмы ****;
На, вот возьми ее скорей!)
А.С.Пушкин «Евгений Онегин»

В Кремле Президент РФ
Владимир **** Путин встретился ….

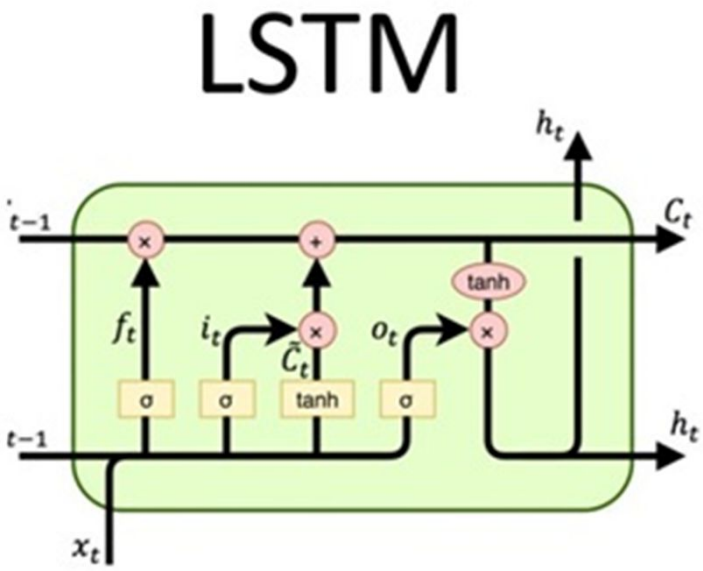
- Предположение, что различие в смысле разных слов определяется разницей контекстов их употребления

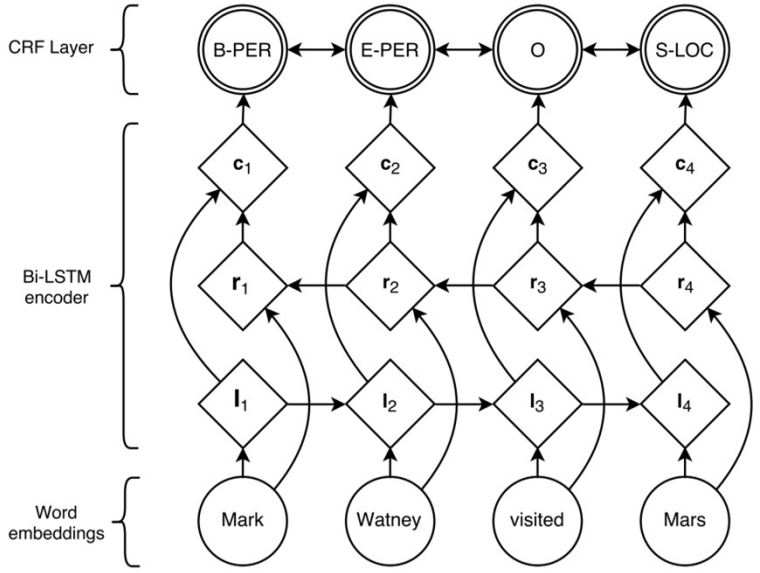
- Дистрибутивные языковые модели, обучаемые на корпусе текстов
  - Word2Vec – векторизация пространства контекстов слова, выбираемая (обучаемая) так, что близкие по смыслу слова должны быть близки, а не связанные – далеки (размерность ~ 300-500)
  - Поиск слова наиболее близкого к точке в пространстве embeddings

KING + (WOMAN – MAN) = QUEEN



Основные нейросетевые архитектуры для обработки текстов с использование эмбеддингов



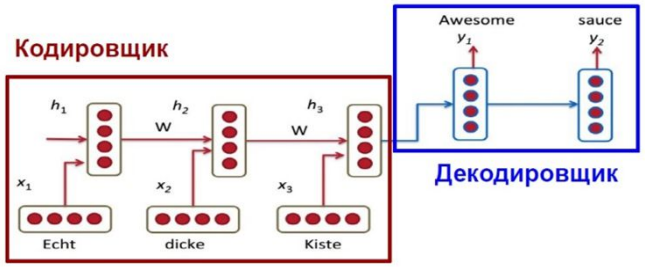


- На вход – векторные представления слов (Word2Vec или аналогичные)
- Для большинства задач обработки текстов лучше подходят реккурентные сети («двигающиеся по тексту»)
- LSTM – (Long Short-Term Memory) - долгая краткосрочная память – взвешивание новой и старой информации, забывание

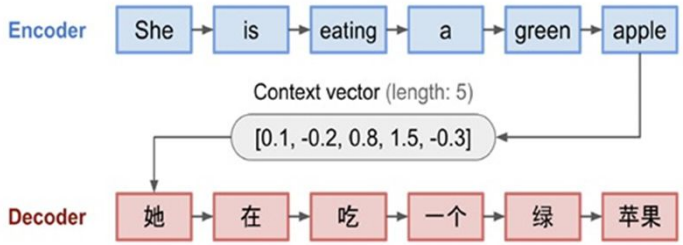
- На практике часто biLSTM + CRF (метод условных случайных полей)

Переход к нейросетевым архитектурам с «вниманием»

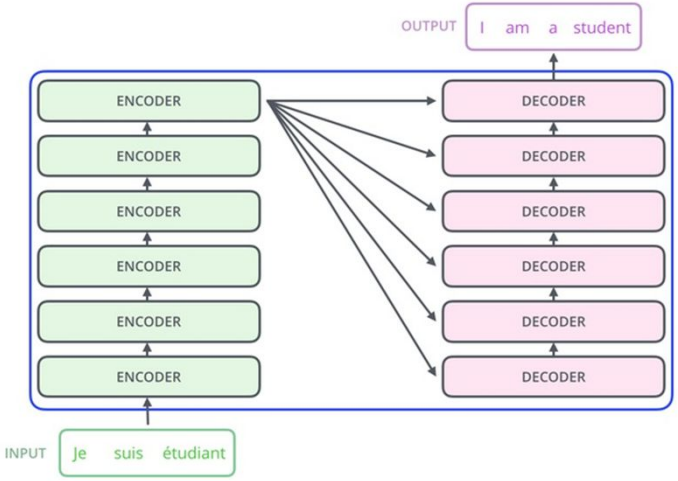
Архитектура seq2seq



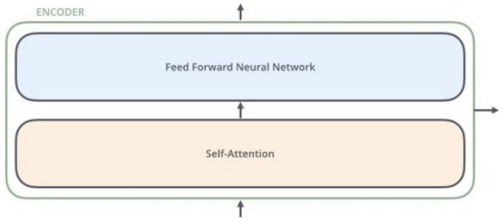
Архитектура seq2seq



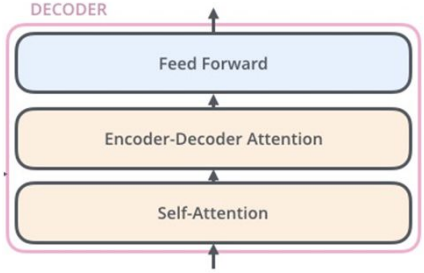
Архитектура «Трансформер»



Encoder



Decoder

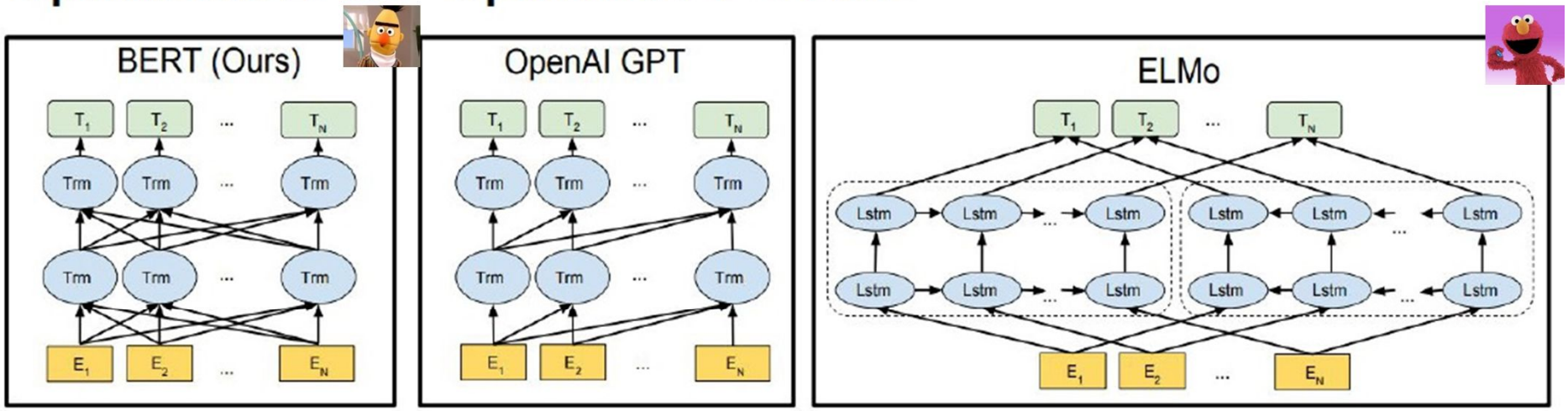


Jay Alammar. The Illustrated Transformer (http://jalammar.github.io/illustrated-transformer/)

И.А.Мажаров «Нейронные сети с механизмом внимания»

Революция 2018 года

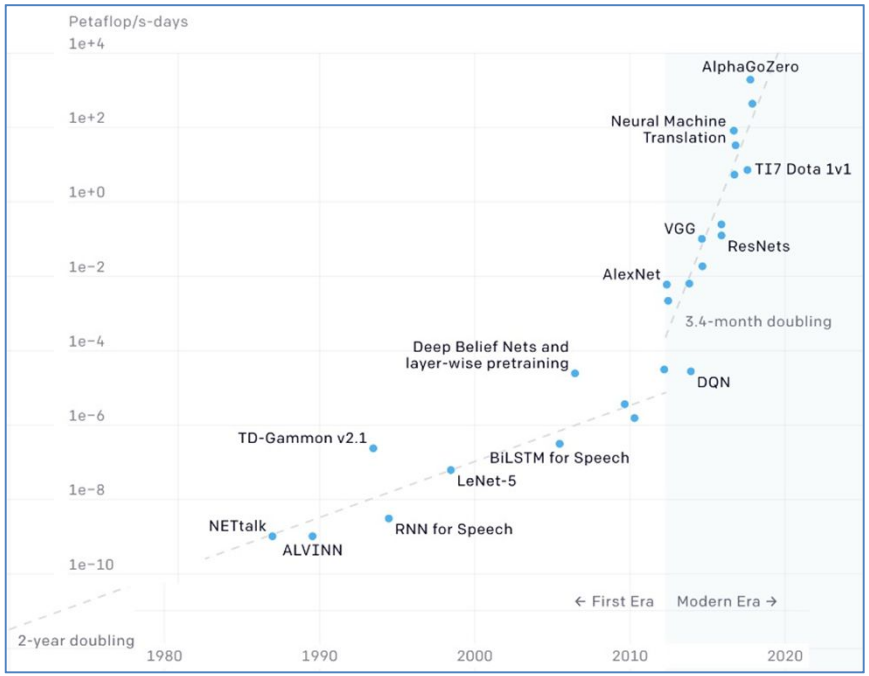
Сравнение BERT - Open AI GPT - ELMo



- BERT использует двунаправленный Трансформер
- Open AI GPT использует left-to-right Трансформер
- ELMo использует конкатенацию результатов двух независимо обученных LSTM (left-to-right и right-to-left)

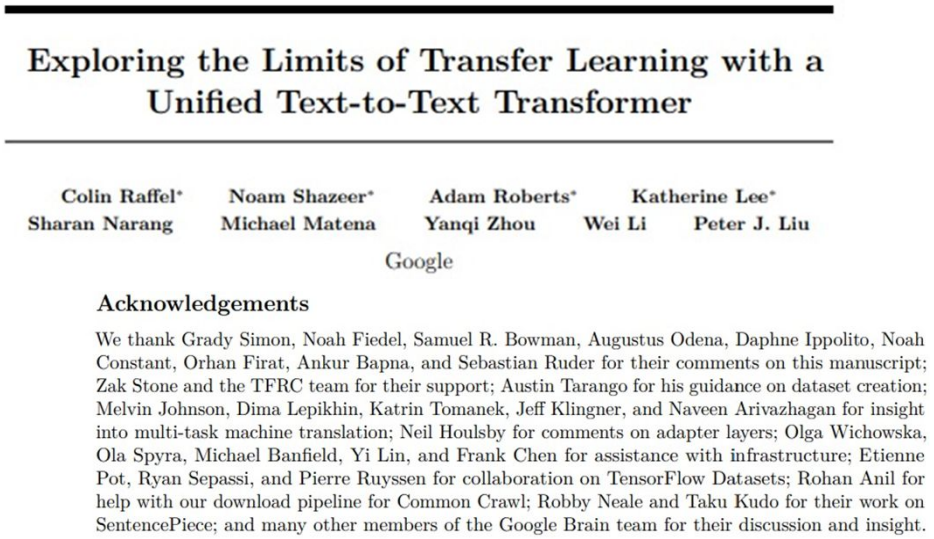
Инфраструктура для обучения современных нейросетевых архитектур

OpenAI. AI and Compute
https://openai.com/blog/ai-and-compute/



Зависимость общих вычислительных затрат для обучения ключевых нейросетевых архитектур

Raffel C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer //arXiv preprint arXiv:1910.10683. – 2019



Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel* Noam Shazeer* Adam Roberts* Katherine Lee*
Sharan Narang Michael Matena Yanqi Zhou Wei Li Peter J. Liu

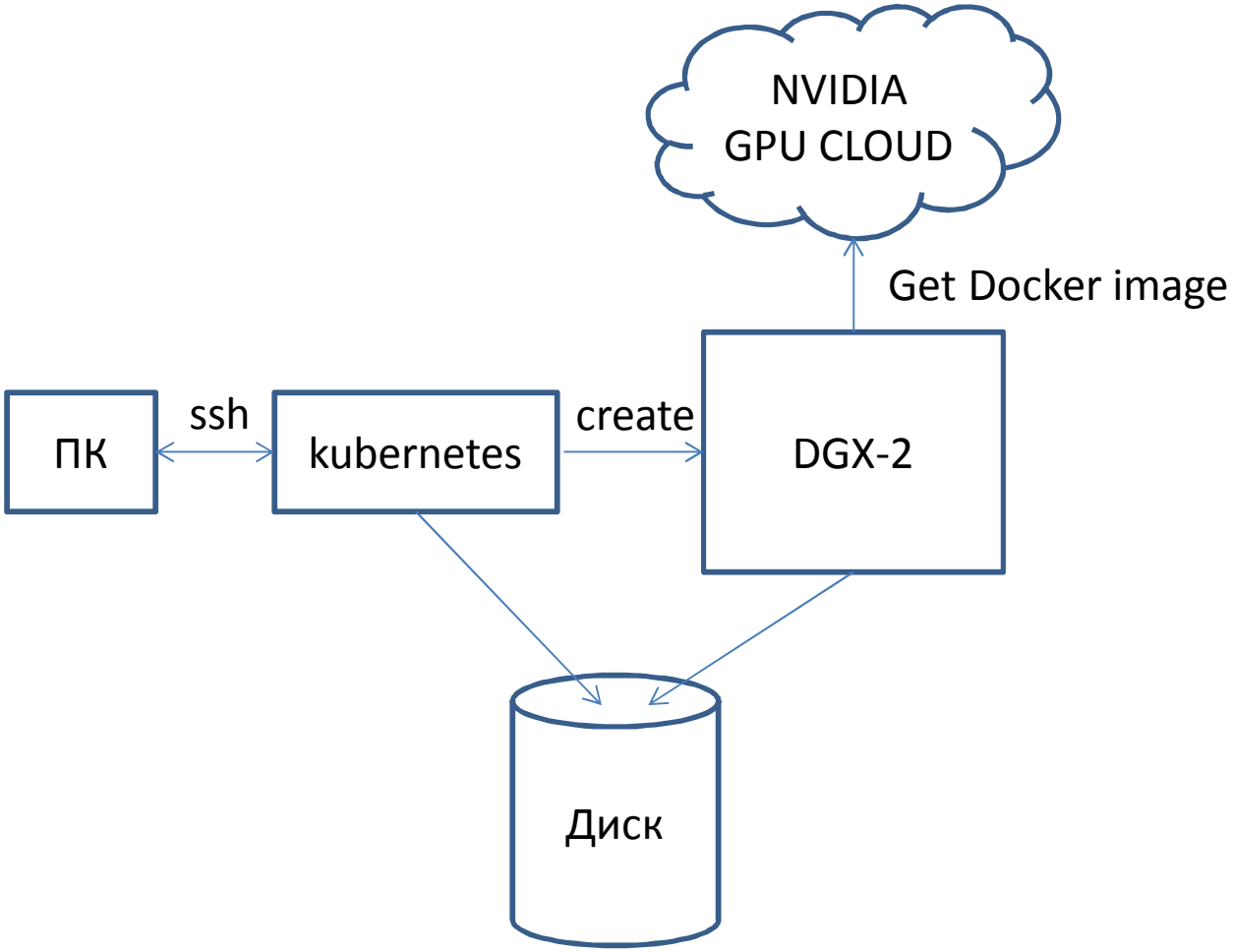
Google

Acknowledgements

We thank Grady Simon, Noah Fiedel, Samuel R. Bowman, Augustus Odena, Daphne Ippolito, Noah Constant, Orhan Firat, Ankur Bapna, and Sebastian Ruder for their comments on this manuscript; Zak Stone and the TFRC team for their support; Austin Tarango for his guidance on dataset creation; Melvin Johnson, Dima Lepikhin, Katrin Tomanek, Jeff Klingner, and Naveen Arivazhagan for insight into multi-task machine translation; Neil Houlsby for comments on adapter layers; Olga Wichowska, Ola Spyra, Michael Banfield, Yi Lin, and Frank Chen for assistance with infrastructure; Etienne Pot, Ryan Sepassi, and Pierre Ruyssen for collaboration on TensorFlow Datasets; Rohan Anil for help with our download pipeline for Common Crawl; Robby Neale and Taku Kudo for their work on SentencePiece; and many other members of the Google Brain team for their discussion and insight.

TPU v3 pod (1024 TPU v3),
примерно в 50 раз быстрее DGX-2
на задачах глубокого обучения

Опыт использования DGX-2
Архитектура взаимодействия



Шаги взаимодействия

- Зайти на сервер по ssh
- Перенести на общее хранилище необходимые код / данные (scp)
- Создать yaml конфигурационный файл в котором необходимо описать:
  - Docker image, который будет использоваться
  - Последовательность команд для выполнения
  - Используемые ресурсы (количество gpu), опционально
  - Указать директорию, которая будет видна в контейнере
- Запустить через kubectl create

Пример yaml файла

```
... (текст файла указан не полностью)
spec:
  nodeName: cngpu01
  imagePullSecrets:
  - name: nvcr.dgxkey
  containers:
  - name: cuda-container
    image: nvcr.io/nvidia/pytorch:19.12-py3
    imagePullPolicy: IfNotPresent
    command: ["/bin/bash","-c", "cd ./pretrain_bert && ./run_train_distributed.sh"]
    volumeMounts:
    - mountPath: /workdir
      name: test-volume
    resources:
      limits:
        nvidia.com/gpu: 16
volumes:
  - name: test-volume
    hostPath:
      # directory location on host
      path: /data
      # this field is optional
      type: Directory
  restartPolicy: Never
```

Загружается из NGC

Если необходимы дополнительные пакеты, которых нет в docker image, то нужны команды для установки

Если свободных GPU в таком количестве нет, задача не будет запущена

Нынешние проблемы

- Нет внутреннего, пользовательского репозитория для docker контейнеров
- Отсутствие поддержки очереди задач
- Если не указывать limit gpu – видны все gpu, если указывать, то как посмотреть состояние gpu для своих запущенных процессов
- Стабильность работы DGX-2 (периодическое отключение, что негативно сказывается на длительных процессах обучения)

В связи с некоторыми из этих проблем планируется переход на singularity + slurm.

Типы задач для DGX-2

1. Большая модель, много данных
   - Требуется много gpu (> 8)
   - Время обучения – дни / недели
   - Пример: предобучение моделей, BERT, GPT-2, T5 и других
2. Большая модель, мало данных
   - Требуется мало gpu (1-2)
   - Время обучения – часы
   - Пример: использование больших предобученных моделей для решения конкретных задач, таких как рубрикация, анализ тональности, извлечение именованных сущностей, вопросно-ответные системы …

Опыт использования DGX-2

При работе с DGX-2 необходимо:

- Использовать mixed-precision вычисления (существенное ускорение и экономия видеопамяти)
- Эффективно использовать несколько gpu (distributed training) и нагружать их на максимум (volatile gpu utilization -> 100%)
- Отслеживать процесс обучения
- Иметь возможность восстановить процесс обучения с чекпоинта

Проведенные эксперименты

- Использовался фреймворк pytorch
  - Поддержка fp16 и mixed-precision (nvidia-apex)
  - Поддержка DistributedDataParallel
  - Поддержка современных версий CUDA (10.2)
- Дообучение модели BERT (110m параметров) на 8 миллионах русскоязычных новостей и на 500 тысячах новостей по информационной безопасности
  - Используется в множестве задач, таких как NER, sentiment analysis, classification
- Дообучение GPT-2 (335m параметров) на 8 миллионах русскоязычных новостей
  - Может использоваться в таких задачах как аннотирование, вопросно-ответные системы, генерация
- Распознавание именованных сущностей в области информационной безопасности с использованием BERT

Проведенные эксперименты-2

- Использование дообученного на предметную область BERT для задачи извлечения именованных сущностей дало существенный прирост в качестве по сравнению с версией без дообучения.
- Процесс дообучения на 500 тысячах новостей = двое суток

Label	RuBERT F1 span	RuCyBERT F1 span
DEVICE	0.429	**0.535**
EVENT	0.662	**0.688**
HACKER	0.589	**0.684**
LOC	**0.911**	**0.911**
ORG	0.792	**0.808**
PER	0.838	**0.863**
PROGRAM	0.654	**0.683**
TECH	0.673	**0.712**
VIRUS	0.459	**0.613**
F1-micro	0.718	**0.752**
F1-macro	0.667	**0.723**

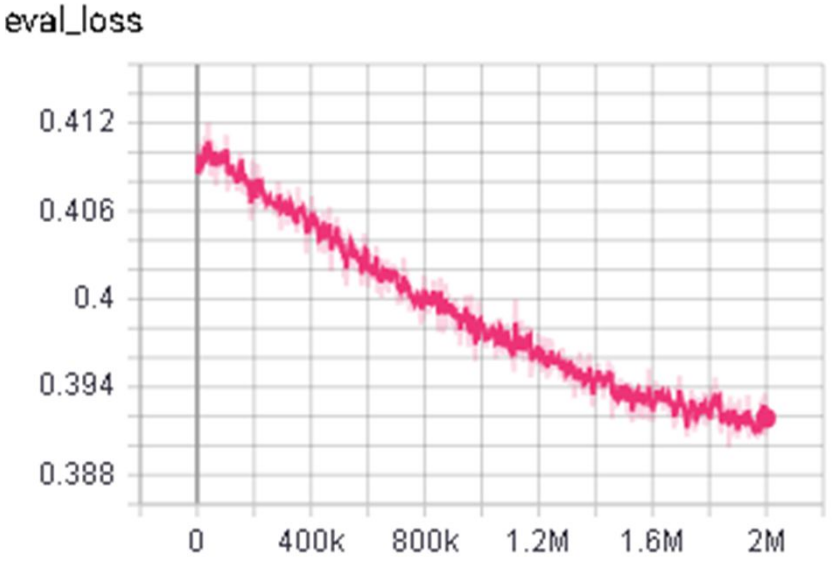
Подготовлены три статьях, принятые к публикации.

Количество запусков в одной конфигурации измеряется в сотнях, а каждый запуск часы.

Расчет такого количества экспериментов в разумные сроки на одной видеокарте не представляется возможным.

Проведенные эксперименты-2

- Дообучение BERT на 8 миллионах новостях на протяжении 4 миллионов шагов.
- Этот процесс занял 34 дня на DGX-2.
- Полная сходимость пока не была достигнута (на изображении последние 2 миллиона шагов).



Данная модель будет использована для задач обработки естественного языка на новостях, таких как аннотирование, классификация и другие.

Проведенные эксперименты-3

Языковая модель GPT-2, дообучалась на 8 миллионах новостях на протяжении 2.5 миллионах шагов.

«**За последние сутки в России зафиксировали 8946 новых случаев заражения коронавирусом.** Об этом сообщает официальный сайт Роспотребнадзора . В ведомстве отметили , что за аналогичный период прошлого года было зарегистрировано 8920 случаев . В целом по России за последнюю неделю зарегистрировано почти два миллиона случаев заболевания . Больше всего пострадавших в Москве , Московской , Тульской , Ростовской областях , Санкт - Петербурге и Хабаровском крае .»

Задача извлечения именованных сущностей

- Именованные сущности используются в большом количестве приложений
- Наиболее часто извлекаемые именованные сущности: Имя человека, Имя организации, Имя локации
  - Качество извлечения на основе методов машинного обучения достигает выше 95%
- В конкретных предметных областях необходимо извлекать другие типы сущностей
  - Необходима разметка обучающей выборки
  - Качество обычно существенно ниже
- Задача: извлечение именованных сущностей в области компьютерной безопасности
  - Необходимо максимально быстро получать актуальную информацию об уязвимостях, вирусах, хакерской активности.

NER в области информационной безопасности: корпус-2

Набор тегов для ручной разметки

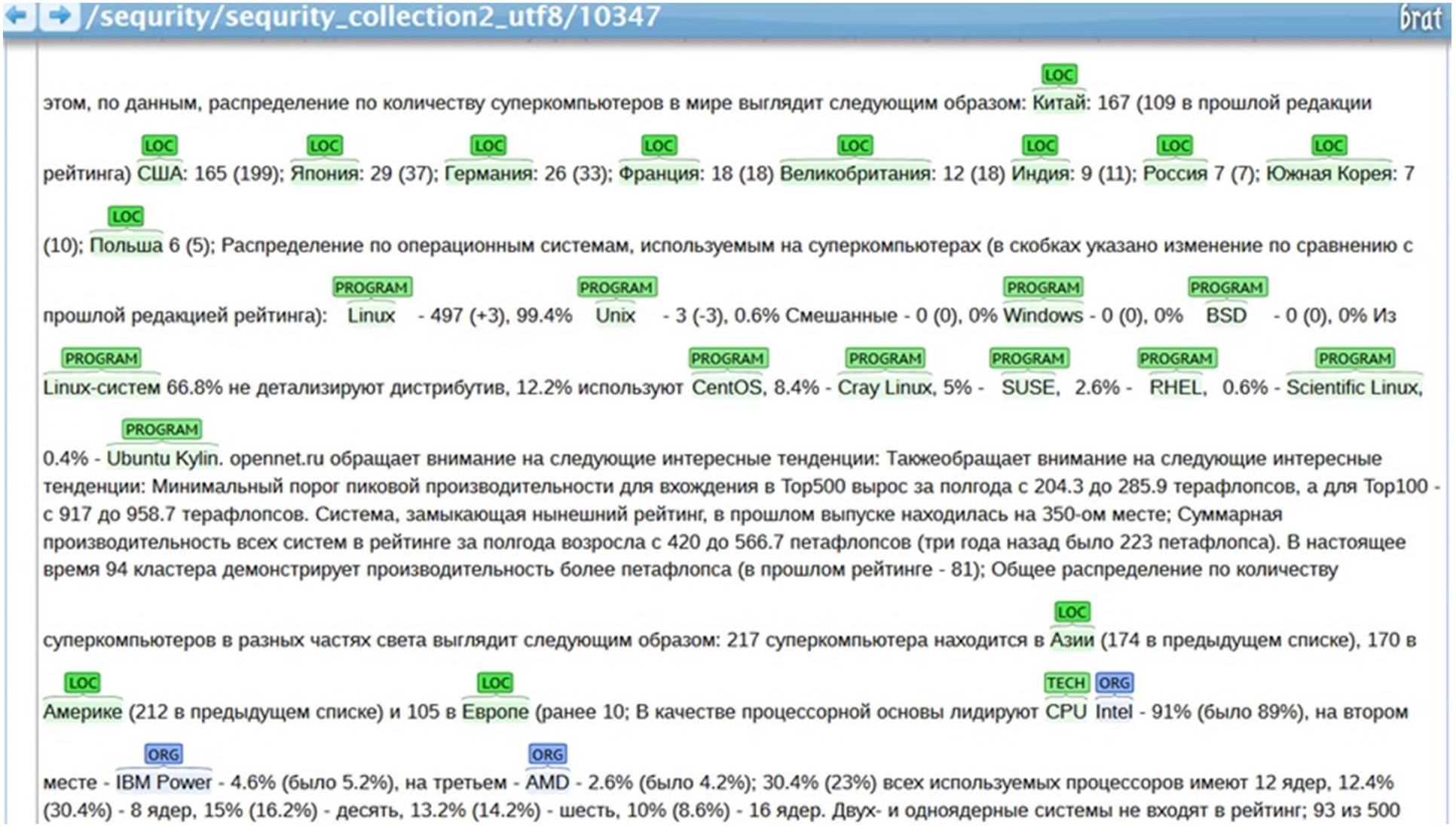
- **Hacker** - отдельные хакеры, группы хакеров;
- **Program** - программы, в том числе сайты, функции, части программ;
- **Device** - электронное оборудование;
- **Tech** - технологии, написанные с большой буквы;
- **Virus** - зловредное ПО разной природы;
- **Event** - различные события и мероприятия.

Теги, приписываемые автоматически

- **Person**
- **Loc**
- **Org**

Итоговый объем корпуса - 861 текст (406488 токенов)

NER в области информационной безопасности: пример разметки (BRAT)



NER в области информационной безопасности: BERT

Использование современных нейросетевых подходов показывают лучшие результаты на задачах подобного типа. Одной из последних архитектур является BERT.

1. Многослойная нейронная сеть на основе архитектуры трансформера
2. Для каждого токена текста формирует зависимое от контекста векторное представление специальным образом
3. Предобучается на гигабайтах данных на задаче предсказания замаскированного слова
4. В базовом варианте 110m параметров

Предобученные модели BERT

Предобученные варианты, работающие на русском языке:

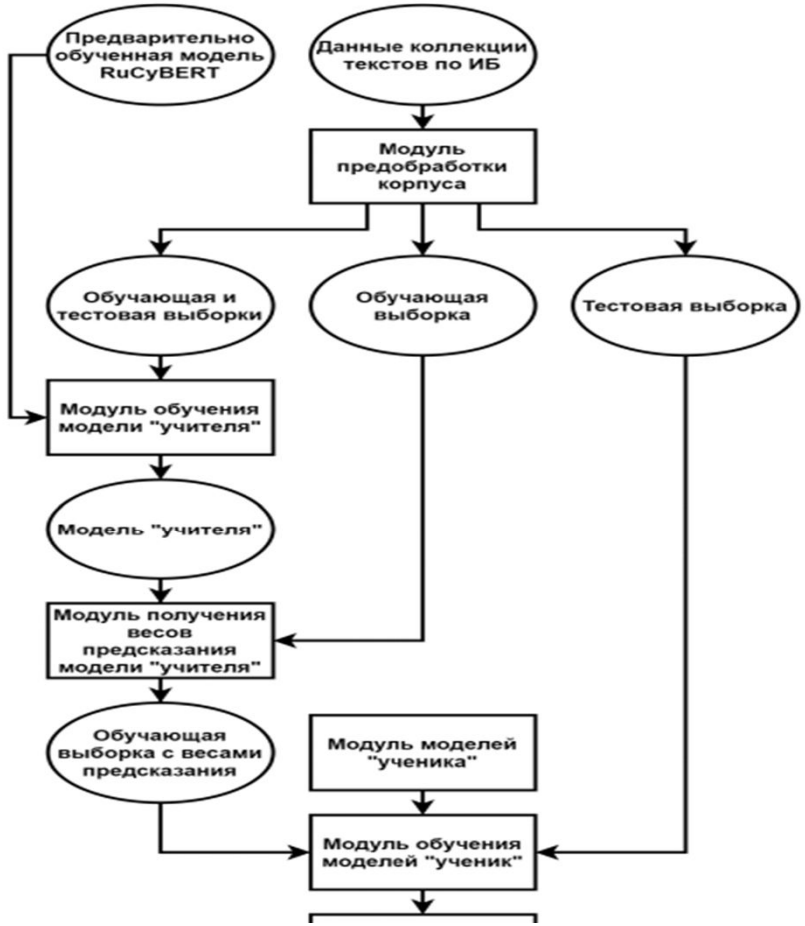
1. Мультиязычный BERT (от авторов архитектуры)
2. RuBERT – дообученная мультиязычная версия с измененным словарем под русские тексты. Показала улучшение качества на наборе задач на русском языке по сравнению с мультиязычной версией
3. Было осуществлено дообучение RuBERT на 500 т. текстах по информационной безопасности - RuCyBERT

NER в области информационной безопасности: результаты

	CRF	BERT	RuBERT	RuCyBERT
DEVICE	31.78	34.04	42.96	**53.52**
EVENT	42.7	60.38	66.19	**68.82**
HACKER	26.58	42.69	58.89	**68.43**
LOC	82.3	90	91.09	**91.1**
ORG	68.15	76.1	79.27	**80.8**7
PER	67.1	80.99	83.85	**86.38**
PROGRAM	62.15	63.15	65.45	**68.31**
TECH	60.65	67.08	67.34	**71.21**
VIRUS	40.9	40.21	45.94	**61.39**
F-micro	63.95	69.37	71.79	**75.21**
F-macro	53.59	61.63	66.77	**72.34**

Магистерская диссертация «Упрощение моделей нейронных сетей для задачи выделения именованных сущностей» Мажаров И.А. (625 группа ВМиК)

	RuCyBERT			BiLSTM			$BiLSTM_{distill}$			BiLSTM-CRF			$BiLSTM\text{-}CRF_{distill}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Total micro	74,65	78,24	77,34	63,50	64,89	64,19	70,28	71,31	70,79	71,47	64,97	68,07	70,67	69,70	70,18
Total macro	72,81	80,2[illegible]	75,35	53,88	51,82	52,19	68,14	57,23	60,04	71,14	56,99	61,24	72,04	60,1[illegible]	64,00



- Обучение более простой сети на результатах обучения более сложной
- Позволяет выявить лучшую конфигурацию более простой сети
- Затем обучать уже более простую сеть на других наборах данных

Выводы

- Идет стремительный прогресс в нейросетевых методах решения задач обработки текстов
- Все более сложные архитектуры требуют все больших вычислительных ресурсов
- Высокопроизводительные вычислители, обученные модели становятся стратегическим ресурсом

- Проведена апробация DGX-2 для задач обработки текстов
- Получены модели RuNewsBERT, RuCybBERT, newsGPT
- Применение обученных моделей позволило улучшить результаты в залачах поиска редких типов именованных сущностей, формирование абстрагированных аннотаций

Спасибо за внимание.

Вопросы?