

# Методы оценки сложности текстовых данных для ускорения обучения языковых моделей с помощью обучения по плану

Сурков Максим Константинович  
Научный руководитель: Ямщиков Иван Павлович

Санкт-Петербургская школа физико-математических и компьютерных наук  
НИУ ВШЭ СПб

9 июня 2021 г.

- Основные задачи NLP и их приложения
  - 1 классификация текстов (спам, грубая речь в соц. сетях)
  - 2 машинный перевод (яндекс.переводчик)
  - 3 построение вопросно-ответных систем (чат-боты)
- Как решаются задачи в NLP
  - 1 Раньше: небольшие языковые модели
  - 2 Сейчас: трансформеры (большие нейронные сети со сложной архитектурой)
- В данной работе используется трансформер BERT
  - 1 наиболее популярный
  - 2 имеет высокое качество
  - 3 имеет сравнительно небольшой размер  $\Rightarrow$  удобно ставить эксперименты

# Мотивация. Обучение языковой модели

- Для применения модели ее нужно обучить
- Обучение состоит из двух этапов:

Этап	Время обучения	Корпус данных	Размер
Предобучение	1-2 недели	Wikipedia BooksCorpus	3-600M 74M
Дообучение	1-2 дня	HND s140 ISWL QQP MNLI	600k-2M 1.6M 200-230k 364k 393k

- Проблемы:
  - **долго** обучать
  - нужно обрабатывать **большие** объемы данных

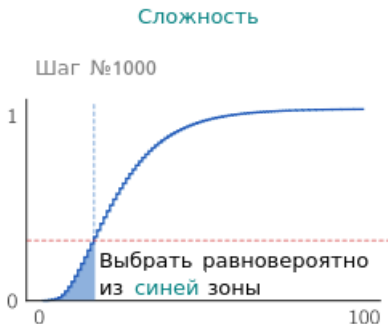
# Обучение по плану. Определение

Обучение по плану состоит из:

- 1 сортировки данных по **метрике** сложности
- 2 семплирования данных (алгоритм выборки данных из датасета)

Пример<sup>1</sup>

- 1 сортируем тексты по длине (метрика=длина)
- 2 семплируем данные из синей зоны
- 3 синяя зона **растет** вправо в течение обучения
- 4 модель учится на все более сложных примерах



<sup>1</sup>Platanios et al., Competence-based Curriculum Learning for Neural Machine Translation, 2019

- ❶ В большинстве работ изучается влияние обучения по плану на задачах
  - Машинного перевода (Platanios et al. (2019), Kosmi et al. (2017))
  - NLU<sup>1</sup> (Xu et al. (2020))но не на
  - **Предобучении** языковых моделей
  - **Классификации** текстов
- ❷ Обычно берутся очевидные метрики (например, длина)
  - нет работ, которые расширяют множество метрик
- ❸ Подавляющее большинство статей использует обучение по плану для улучшения **качества** модели, но **не скорости** ее обучения

---

<sup>1</sup>Построение вопросно-ответных систем

**Цель:** исследовать возможность ускорения обучения языковой модели BERT на задачах предобучения модели и классификации текстов с помощью обучения по плану за счет применения улучшенной метрики сложности текстовых данных.

**Задачи:**

- 1 Предложить метрики оценки сложности текста
- 2 Реализовать производительные алгоритмы вычисления предложенных метрик на больших корпусах данных
- 3 Сравнить эффективность найденных метрик
- 4 Исследовать влияние найденных метрик на скорость обучения языковой модели BERT на разных типах тренировочных данных

Метрика	Обозначение	Ссылка
Длина	длина	Platanios et al., 2019
вероятность правдоподобия	правд.	Platanios et al., 2019
ранг самого редкого слова в тексте	ранг	Xuan Zhang et al., 2018
$L_1$ норма вектора TF-IDF	TF-IDF	Эта работа
Excess Entropy	EE	адаптация Nihat Ay et al., 2006
TSE	TSE	адаптация Nihat Ay et al., 2006
модельная	MLM-loss	Эта работа
Среднее число токенов в слове	TPW	Эта работа

- СТАТИСТИКИ

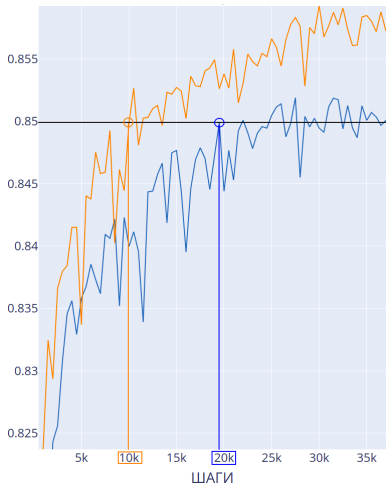
- ① длина  $\rightarrow$  число текстов с такой длиной
  - ②  $(i, x_i) \rightarrow$  число текстов, где  $t_i = x_i$
  - ③  $(x_i) \rightarrow$  число текстов, где  $x_i$  является последним токеном
  - ④  $(i, x_{i-1}, x_i) \rightarrow$  число текстов, где на  $(i-1)$ -й позиции стоит  $x_{i-1}$ , а на  $i$ -й позиции стоит  $x_i$
  - ⑤  $x_i \rightarrow$  число текстов, в которых есть  $x_i$
- сбор статистик в параллельном режиме (разделение по данным)

Режим	Время
1 CPU	$\approx 2$ недели
5 CPU	$\approx 1-2$ дня
20 CPU	$< 14$ ч.
40 CPU	$< 6$ ч.



# Метод сравнения алгоритмов обучения

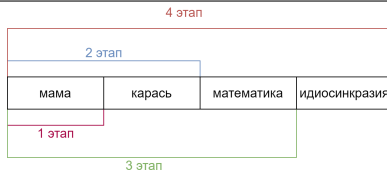
- Метрика – объект изучения
- Хочется понять, как разные метрики влияют на скорость обучения
- Для этого нужно научиться сравнивать две метрики
- Для этого:
  - 1 фиксируем все (кроме метрик)
  - 2 сравниваем среднее число шагов, необходимое для достижения порога



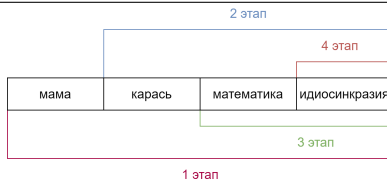
# Семплеры

В данной работе использованы следующие семплеры:

CB (Platanious et al., 2019) -  
префиксный семплер



DB - суффиксный семплер



Нур - оконный семплер



В данной работе использованы следующие семплеры:

- SS (shuffle-sort)
  - 1 случайно поделим на батчи
  - 2 отсортируем батчи по средней сложности
- SM (sort-merge)
  - 1 отсортируем по длине
  - 2 разобьем на группы
  - 3 каждую группу отсортируем по сложности
  - 4 строим батч из примеров каждой группы

# Сравнение метрик: предобучение

Датасет: BooksCorpus

Метрика	Порог	Семплы					min loss
		CB	DB	Нур	SS	SM	
ранг	2.00	$\infty$	17.5k	<b>16.5k</b>	<b>16.5k</b>	27k	<b>1.58</b>
TF-IDF	2.00	$\infty$	34k	35k	37.5k	$\infty$	1.84
EE	3.50	$\infty$	4k	3.5k	4.5k	9.5k	2.25
TSE	3.50	$\infty$	9k	9k	8.5k	18k	2.60
правд.	3.50	$\infty$	13.5k	13.5k	15.5k	50k	2.83
длина	3.50	$\infty$	50.5k	$\infty$	-	-	3.45
база	2.00	9.5k					1.58

- лучшая метрика – максимальный ранг слова (замедляет в 2 раза без потери качества)
- обучение по плану замедляет обучение от 2 до 5 раз и ухудшает качество модели

стандартное отклонение  $\delta \leq 3k$  шагов

# Сравнение метрик: классификация текстов

Датасеты: sentiment140 (результаты в таблице), HND

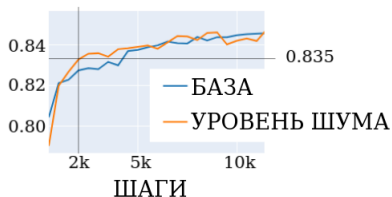
Метрика	Порог	Семплы					Точность
		CB	DB	Нур	SS	SM	
ранг	85.5%	70k	18.5k	19.5k	<b>17k</b>	19k	86.7%
TF-IDF	85.5%	115.5k	21.5k	19.5k	<b>16.5k</b>	22k	86.7%
EE	85.5%	59k	19.3k	23k	20k	19k	86.7%
TSE	85.5%	95.5k	<b>16.5k</b>	20.5k	21.5k	18k	86.8%
правд.	85.5%	112k	17.5k	21.5k	17.5k	21.5k	86.7%
длина	85.5%	112.5k	20k	19k	-	-	<b>86.2%</b>
MLM-loss	85.5%	59.5k	21k	23.5k	19.5k	20k	<b>86.1%</b>
база	85.5%	17.5k					87%

- лучшая конфигурация (TF-IDF+SS) ускоряет обучение до 3% в среднем в сравнении с базой
- длина и MLM-loss уменьшают точность модели на 0.6%
- в общем случае нет значительного ускорения обучения

# Влияние метрик на скорость обучения. Шум

- Wu et al., When Do Curricula Work?, 2020 – аналог для CV
- Для рассмотрения данного частного случая нужно искусственно добавить шум в данные
  - 1 выберем  $p \sim U[0, 0.4]$  – уровень шума
  - 2 применим один из трех видов шумов к  $p$  буквам в тексте
  - 3 виды шума:
    - 1 **клавиатурный** – замена на случайного соседа по клавиатуре (Kumar et al. (2020)))
    - 2 ошибки произношения
    - 3 случайная перестановка двух символов в слове
- Оказалось, что метрика "уровень шума" ускоряет обучение до 2.15 раз на старте (семплер: DB, задача классификации текстов)

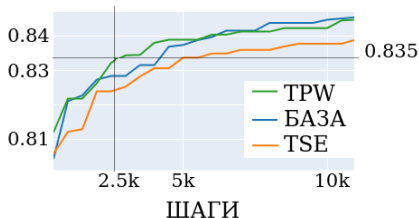
Метрики	Порог	Шаги
уровень шума	83.5%	2k
база	83.5%	4.3k



# Влияние метрик на скорость обучения. Шум

- В реальности мы не обладаем информацией о количестве шума в конкретном примере  $\Rightarrow$  нужно придумать метрику такую, что:
  - 1 сильно коррелирует с уровнем шума
  - 2 не опирается на информацию о шуме
- Выяснилось, что подходит метрика TPW (среднее число токенов на слово)
  - у шумных данных TPW больше  $\Rightarrow$  модель сначала учится на чистых примерах, плавно переходя к более шумным
- TPW ускоряет обучение в 1.72 раза для достижения 95% (83.5% из 87%) итоговой точности (сеплер: DB, задача классификации текстов)

Метрики	Порог	Шаги
TSE	83.5%	5k
TPW	83.5%	2.5k
база	83.5%	4.3k



- ❶ Предложен широкий спектр метрик оценки сложности текста
  - метрики TSE и EE адаптированы под задачу обработки языка
- ❷ Реализованы производительные алгоритмы подсчета метрик на больших объемах данных
- ❸ Результаты применения обучения по плану на задаче предобучения
  - длина – худшая метрика на предобучении (замедляет обучения до 12 раз, уменьшает качество модели)
  - максимальный ранг слова – лучшая метрика на предобучении (замедляет в 2 раза **без потери качества**)
- ❹ Дообучение на задаче классификации текстов
  - лучшая конфигурация (TF-IDF+SS) ускоряет обучение до 3% в среднем на классификации
  - длина и MLM-loss уменьшают точность модели на 0.6%
- ❺ Шумные тренировочные данные
  - метрика TPW ускоряет обучение в **1.72 раза** для достижения 95% итоговой точности на шумном корпусе данных



- Ay, N., Olbrich, E., Bertschinger, N., & Jost, J. (2006, August). A unifying framework for complexity measures of finite systems. In Proceedings of ECCS (Vol. 6).
- Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009, June). Curriculum learning. In Proceedings of the 26th annual international conference on machine learning (pp. 41-48).
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

- Hacoen, G., & Weinshall, D. (2019, May). On the power of curriculum learning in training deep networks. In International Conference on Machine Learning (pp. 2535-2544). PMLR.
- Kocmi, T., & Bojar, O. (2017). Curriculum learning and minibatch bucketing in neural machine translation. arXiv preprint arXiv:1707.09533.
- Kurdi, M. Z. (2020). Text Complexity Classification Based on Linguistic Information: Application to Intelligent Tutoring of ESL. arXiv preprint arXiv:2001.01863.
- Mermer, M. N., & Amasyali, M. F. (2017). Scalable Curriculum Learning for Artificial Neural Networks. IPSI BGD TRANSACTIONS ON INTERNET RESEARCH, 13(2).

- Narasimhan, S., Narasimhan, V. A. P. B. S., Karch, G., Rao, R., Huang, J., Zhang, Y., Ginsburg, B., Chitale, P., Sreenivas, S., Mandava, S., Ginsburg, B., Forster, C., Mani, R., & Kersten, K. (2020, October 13). NVIDIA Clocks World's Fastest BERT Training Time and Largest Transformer Based Model, Paving Path For Advanced Conversational AI. NVIDIA Developer Blog. <https://developer.nvidia.com/blog/training-bert-with-gpus/>
- Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B., & Mitchell, T. M. (2019). Competence-based curriculum learning for neural machine translation. arXiv preprint arXiv:1903.09848.
- Sajjad, H., Dalvi, F., Durrani, N., & Nakov, P. (2020). Poor Man's BERT: Smaller and Faster Transformer Models. arXiv preprint arXiv:2004.03844.

- Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M. W., & Keutzer, K. (2020). Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 8815–8821. <https://doi.org/10.1609/aaai.v34i05.6409>
- van der Sluis, F., & van den Broek, E. L. (2010, August). Using complexity measures in information retrieval. In Proceedings of the third symposium on information interaction in context (pp. 383-388).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.

- Xu, B., Zhang, L., Mao, Z., Wang, Q., Xie, H., & Zhang, Y. (2020). Curriculum Learning for Natural Language Understanding. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 6095–6104.  
<https://doi.org/10.18653/v1/2020.acl-main.542>
- Zhang, X., Kumar, G., Khayrallah, H., Murray, K., Gwinnup, J., Martindale, M. J., ... & Carpuat, M. (2018). An empirical exploration of curriculum learning for neural machine translation. arXiv preprint arXiv:1811.00739.

# Дополнительно: Поиск метрик

метрика	формула
Мультиинформация	$\sum_{v \in V} H_p(X_v) - H_p(X_V)$
Избыточная энтропия (ЕЕ)	$\left[ \sum_{v \in V} H(X_{V \setminus \{v\}}) \right] - (n - 1)H(X_V)$
<b>TSE</b>	$\sum_{k=1}^{n-1} \frac{k}{n} C^{(k)}(X_V), \text{ где}$ $C^{(k)}(X_V) =$ $\frac{n}{k \binom{n}{k}} \sum_{A \subseteq V,  A =k} H(X_A) - H(X_V)$
Переходная информация	:(

$$V = \{1, \dots, n\}, X_V = (X_1, \dots, X_n)$$

Nihat Ay et al., A **Unifying** Framework for Complexity Measures of Finite Systems, 2006

# Дополнительно: Адаптация ЕЕ и TSE под задачи обработки языка

## 1 Образование совместной случайной величины

$$T = (t_1, t_2, \dots, t_{i-1}, t_i, \dots, t_n)$$

$t_i \rightarrow \xi_{t_i}^i =: \mu_i$  – бинарная случайная величина



$$\xi = (\xi_{t_1}^1, \xi_{t_2}^2, \dots, \xi_{t_{i-1}}^{i-1}, \xi_{t_i}^i, \dots, \xi_{t_n}^n)$$

## 2 Вычисление энтропии

$$H(\mu) = \sum_{i=1}^n H(\mu_i | \mu_1, \mu_2, \dots, \mu_{i-1}) = \sum_{i=1}^n H(\mu_i | \mu_{i-L}, \dots, \mu_{i-1})$$

## 3 $L = 1$

$$H(\mu) = H(\mu_1) + H(\mu_2 | \mu_1) + \dots + H(\mu_i | \mu_{i-1}) + \dots + H(\mu_n | \mu_{n-1})$$

$$EE(X) = \left[ \sum_{v \in V} H(X_{V \setminus \{v\}}) \right] - (n-1)H(X_V) =$$

$$\left[ \sum_{i=1}^n H(\mu_1, \dots, \mu_{i-1}, \mu_{i+1}, \dots, \mu_n) \right] - (n-1)H(\mu)$$

- $\mathcal{O}(n^2)$
- $\mathcal{O}(n)$

$$\sum_{i=1}^n H(\mu_1, \dots, \mu_{i-1}, \mu_{i+1}, \dots, \mu_n) =$$

$$= \sum_{i=1}^n H(\mu) - H(\mu_i | \mu_{i-1}) - H(\mu_{i+1} | \mu_i) + H(\mu_{i+1})$$

$$EE(X) = \sum_{i=2}^n H(\mu_i) - H(\mu_i | \mu_{i-1}) = \sum_{i=2}^n I(\mu_{i-1} : \mu_i)$$



$$\sum_{k=1}^{n-1} \frac{k}{n} C^{(k)}(X_V)$$
$$C^{(k)}(X_V) = \frac{n}{k \binom{n}{k}} \sum_{A \subseteq V, |A|=k} H(X_A) - H(X_V) =$$
$$= \frac{n}{k} \left[ \frac{1}{\binom{n}{k}} \sum_{A \subseteq V, |A|=k} H(X_A) \right] - H(X_V)$$

## Дополнительно: Вычисление TSE

$$\frac{1}{\binom{n}{k}} \sum_{A \subseteq V, |A|=k} H(X_A) = \frac{1}{\binom{n}{k}} \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} H(\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_k})$$

- ❶  $\mathcal{O}^*(2^n)$
- ❷  $\mathcal{O}(n^2)$  - динамическое программирование
- ❸  $\mathcal{O}(n)$

$$\sum_{i=1}^n A_i H(\mu_i) + \sum_{i=2}^n B_i H(\mu_i | \mu_{i-1})$$

$$A_i = \begin{cases} \binom{n-2}{k-1} / \binom{n}{k} = \frac{k(n-k)}{n(n-1)}, & i > 1 \\ \binom{n-1}{k-1} / \binom{n}{k} = \frac{k}{n}, & i = 1 \end{cases}$$

$$B_i = \frac{\binom{n-2}{k-2}}{\binom{n}{k}} = \frac{k(k-1)}{n(n-1)}$$

Датасет: Hyperpartisan News Detection

Метрика	Порог	Семплы					Точность
		CB	DB	Нур	SS	SM	
ранг	92.9%	$\infty$	22k	<b>20.5k</b>	22.5k	39k	93.6%
TF-IDF	92.9%	$\infty$	<b>19.5k</b>	24k	23.5k	33k	93.5%
EE	92.9%	71.5k	25.5k	22.5k	<b>19.5k</b>	32.5k	93.8%
TSE	92.9%	56.5k	<b>21k</b>	23k	22k	31k	93.8%
правд.	92.9%	$\infty$	<b>20k</b>	24k	<b>20k</b>	30k	93.8%
длина	92.9%	55k	23k	22.5k	-	-	93.7%
MLM-loss	92.9%	23.5k	<b>18k</b>	23k	24k	20k	<b>93.9%</b>
база	92.9%	22k					93.8%

стандартное отклонение  $\delta \leq 3k$  шагов