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

11 мая 2021 г.

Мотивация. Применения

- Основные задачи NLP и их приложения
  1. классификация текстов (грубая речь в соч. 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

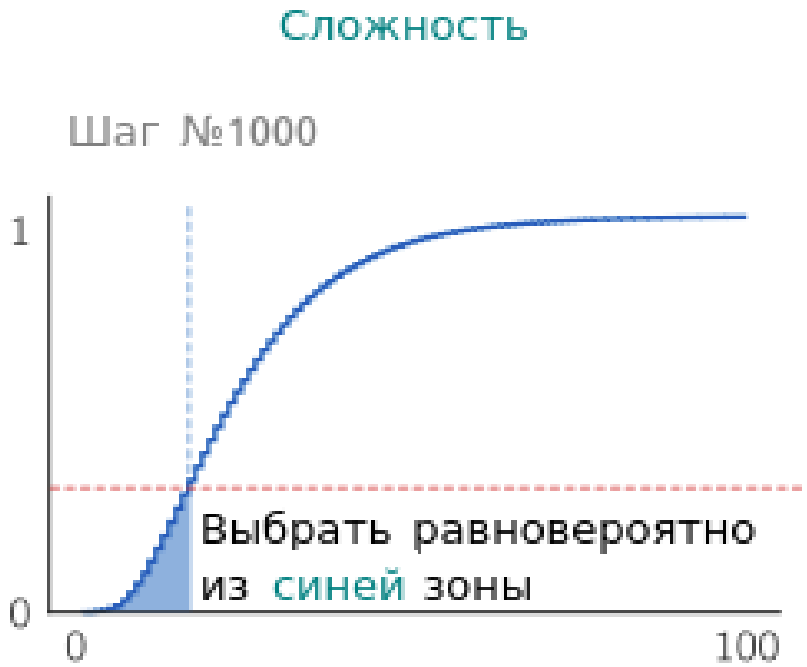
Обучение по плану. Определение

Обучение по плану состоит из:

1. сортировки данных по **метрике** сложности
2. семплирования данных[1]

Пример[2]

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

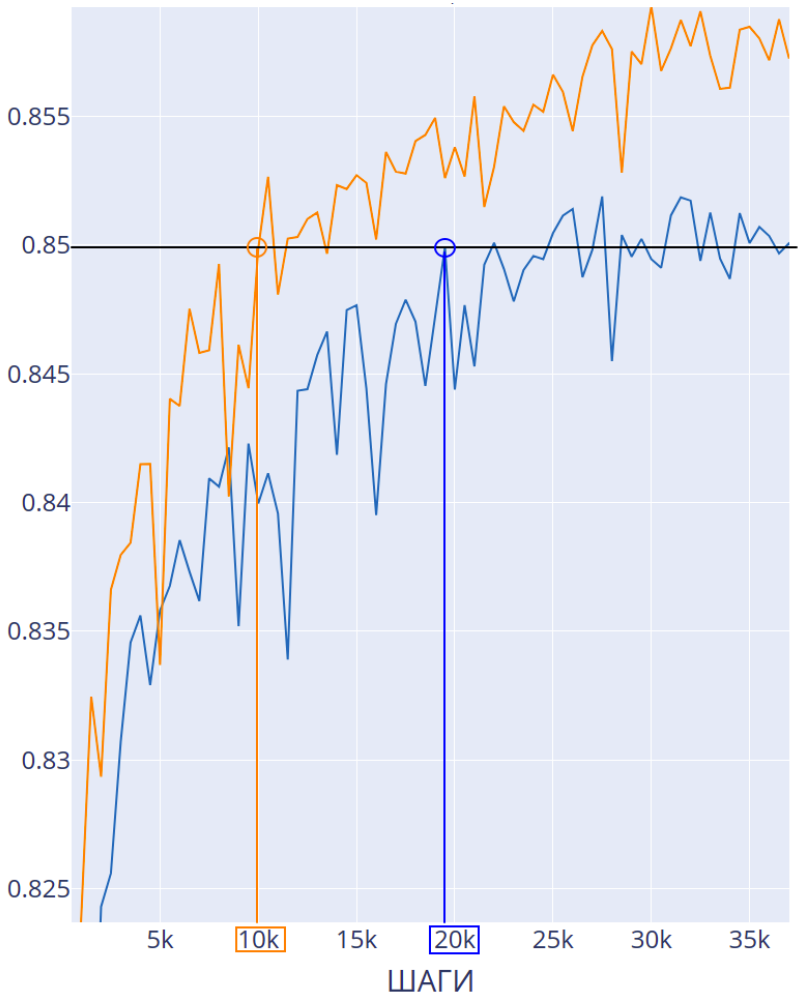


[1] Выбирать примеры из датасета

[2] Platanios et al., Competence-based Curriculum Learning for Neural Machine Translation, 2019

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



Поле исследований

1. Не изучено влияние обучения по плану на задачах **классификации и предобучения**
   - Машинный перевод (Platanios et al. (2019), Kocmi et al. (2017))
   - NLU (Xu et al. (2020))
2. Покрыто **узкое** множество метрик (длина - лучшая метрика на данный момент)
3. Мало исследований влияния обучения по плану на скорость обучения
4. Не рассмотрен **важный** случай с **шумными** тренировочными данными
   - Wu et al. (2020) показали **ускорение** обучения на классификации картинок только при использовании **шумных** датасетов
   - Большинство автоматически собираемых датасетов – шумные
   - Шумные данные дорого очищать

Цель и задачи

Цель: исследовать возможность ускорения обучения языковой модели BERT на задачах классификации и предобучения 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 найденные метрики
4. Исследовать влияние найденных метрик на скорость обучения языковой модели BERT на чистых и шумных тренировочных данных

Поиск метрик

1. база
   - **длина**, вероятность правдоподобия (Platanios et al., 2019)
   - самое редкое слово в предложении (Xuan Zhang et al., 2018)
2. информационный поиск
   - **tf-idf**
3. **теория информации** (Nihat Ay et al., 2006)
   - EE, TSE

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

$$\downarrow$$

$$\xi = (\xi^1_{t_1}, \xi^2_{t_2}, \ldots, \xi^{i-1}_{t_{i-1}}, \xi^i_{t_i}, \ldots, \xi^n_{t_n})$$

Асимптотика	Время
$\mathcal{O}^*(2^n), \mathcal{O}(n^2)$	$>$ 1 мес.
$\mathcal{O}(n)$	$<$ 4ч.

$t_i \to \xi^i_{t_i} =: \mu_i$ — бинарная случайная величина

4. модельная (MLM-loss)
   - учим BERT на задаче MLM (Пример: "Привет, как [МАСКА]?"), оптимизируя кросс-энтропию
   - сложность = значение кросс-энтропии на данном тексте
   - треубет GPU
5. среднее число токенов в слове (TPW)

Вычисление метрик

- 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.

Итого:

- предложены подходы, покрывающие широкое множество метрик
- предложены алгоритмы, вычисляющие метрики за пренебрежимо маленькое время ($< 8\%$ от времени обучения)

Сравнение метрик. Предобучение

$\max \Delta \leq 3k$

Датасет		BooksCorpus					
семплер	Порог	CB	DB	Hyp	SS	SM	min loss
max wf rk	2.00	∞	17.5k	**16.5k**	**16.5k**	27k	**1.58**
TF-IDF	2.00	∞	34k	35k	37.5k	∞	1.84
база	2.00			9.5k			1.58
		BooksCorpus					
EE	3.50	∞	4k	3.5k	4.5k	9.5k	2.25
TSE	3.50	∞	9k	9k	8.5k	18k	2.60
правд.	3.50	∞	13.5k	13.5k	15.5k	50k	2.83
длина	3.50	∞	50.5k	∞	-	-	3.45

- $(\pm)$ 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

Сравнение метрик. Классификация

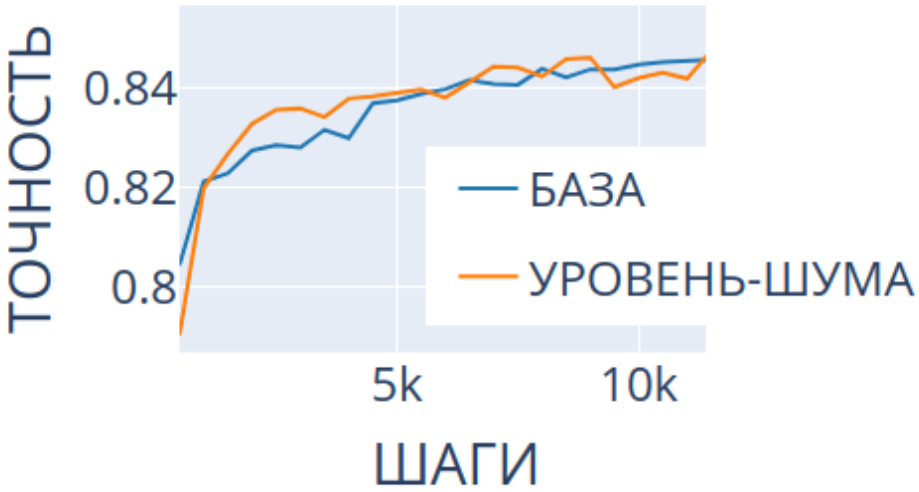
$\max \Delta \leq 3k$

Датасет		sentiment140					
семплер	Порог	CB	DB	Hyp	SS	SM	Точность
длина	85.5%	112.5k	20k	19k	-	-	**86.2%**
TF-IDF	85.5%	115.5k	21.5k	19.5k	**16.5k**	22k	86.7%
TSE	85.5%	95.5k	**16.5k**	20.5k	21.5k	18k	86.8%
EE	85.5%	59k	19.3k	23k	20k	19k	86.7%
max wf rk	85.5%	70k	18.5k	19.5k	**17k**	19k	86.7%
правд.	85.5%	112k	17.5k	21.5k	17.5k	21.5k	86.7%
MLM-loss	85.5%	59.5k	21k	23.5k	19.5k	20k	**86.1%**
база	85.5%			17.5k			87%

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

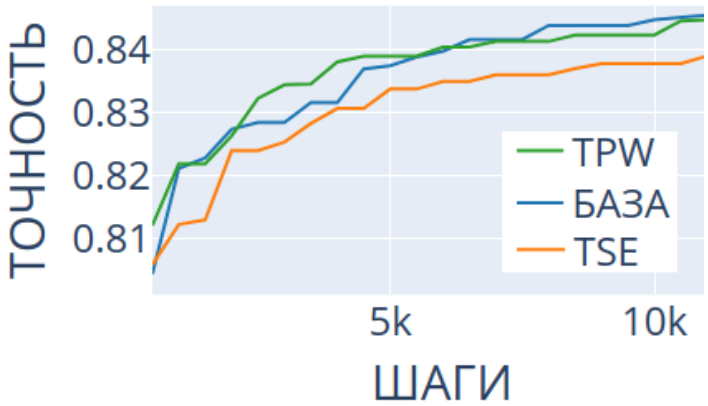
Влияние метрик на скорость обучения. Шум

- 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 2.5



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 2 раза для достижения 95% итоговой точности



Результаты

1. Предложен широкий спектр метрик оценки сложности текста
   - метрики TSE и EE адаптированы под задачу обработки языка
2. Реализованы алгоритмы подсчета метрик на больших объемах данных ($\leq 8\%$ от времени обучения)
3. Предобучение
   - 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**)
4. Классификация
   - лучшая конфигурация (TF-IDF+SS) ускоряет обучение до 3% в среднем на классификации
   - длина и MLM-loss уменьшают точность модели на 0.6%
5. Шумные тренировочные данные
   - метрика TPW ускоряет обучение до **2 раз** для достижения 95% итоговой точности на шумном корпусе данных

Дополнительно: ссылки

- 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.

Дополнительно: ссылки

- Hacohen, 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).

Дополнительно: ссылки

- 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.

Дополнительно: ссылки

- 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.

Дополнительно: ссылки

- 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 (EE)	$\left[\sum_{v \in V} H(X_{V \setminus \{v\}})\right] - (n-1)H(X_V)$
TSE	$\sum_{k=1}^{n-1} \frac{k}{n} C^{(k)}(X_V)$, где $C^{(k)}(X_V) = \frac{n}{k\binom{n}{k}} \sum_{A \subseteq V, \lvert A \rvert = k} H(X_A) - H(X_V)$
Переходная информация	:(

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

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

Дополнительно: Адаптация EE и TSE под задачи обработки языка

1. Образование совместной случайной величины

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

$$t_i \to \xi^i_{t_i} =: \mu_i - \text{бинарная случайная величина}$$

$$\downarrow$$

$$\xi = (\xi^1_{t_1}, \xi^2_{t_2}, \ldots, \xi^{i-1}_{t_{i-1}}, \xi^i_{t_i}, \ldots, \xi^n_{t_n})$$

2. Вычисление энтропии

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

3. $L = 1$

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

Дополнительно: Вычисление EE

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

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

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

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

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

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

Дополнительно: Вычисление TSE

$$\sum_{k=1}^{n-1} \frac{k}{n} C^{(k)}(X_V)$$

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

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

Дополнительно: Вычисление TSE

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

1. $\mathcal{O}^*(2^n)$
2. $\mathcal{O}(n^2)$ - динамическое программирование
3. $\mathcal{O}(n)$

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

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

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

Результаты. Классификация. HND

Корпус данных: Hyperpartisan News Detection
$\max \Delta \leq 3k$

Датасет		HND					
семплер	Порог	CB	DB	Hyp	SS	SM	Точность
длина	92.9%	55k	23k	22.5k	-	-	93.7%
TF-IDF	92.9%	∞	**19.5k**	24k	23.5k	33k	93.5%
TSE	92.9%	56.5k	**21k**	23k	22k	31k	93.8%
EE	92.9%	71.5k	25.5k	22.5k	**19.5k**	32.5k	93.8%
max wf rk	92.9%	∞	22k	**20.5k**	22.5k	39k	93.6%
правд.	92.9%	∞	**20k**	24k	**20k**	30k	93.8%
база	92.9%			22k			93.8%