Автоматическое исправление грамматических ошибок.

Введение.

Алексей Андреевич Сорокин
Yandex Research,
МГУ, отделение теоретической и прикладной лингвистики,
МФТИ, кафедра системных исследований.

Школа РАИИ 2021
5 июля 2021г.

Исправление грамматических ошибок

- Исходное предложение:

  Cars and buses become dengerous and make problem in the street .

- Исправленное предложение:

  Cars and buses become dangerous and cause problems in the street .

- Задача: автоматически генерировать подобные исправления.

Исправление грамматических ошибок

- Исходное предложение:

  Такие знаменитые лица как композиторы Рахманинов и Стравинский и писталь Набоков принадлежают к этой волной .

- Исправленное предложение:

  Такие знаменитости , как композиторы Рахманинов и Стравинский и писатель Набоков , принадлежат к этой волне .

- Задача: автоматически генерировать подобные исправления.

Исправление грамматических ошибок: приложения

Исправление грамматических ошибок: приложения

- Прежде всего задачи обучения иностранному языку.
  - Системы автоматической оценки и проверки.

Исправление грамматических ошибок: приложения

- Прежде всего задачи обучения иностранному языку.
  - Системы автоматической оценки и проверки.
- Автоматические ассистенты и редакторы текста:
  - Проверка орфографической и грамматической корректности (Microsoft Word).
  - Системы улучшения качества текста (Grammarly).
- Облегчение извлечения информации из текста.

Типы грамматических ошибок

- Орфографические ошибки.
- Пунктуационные ошибки.

Типы грамматических ошибок

- Орфографические ошибки.
- Пунктуационные ошибки.
- Неверный выбор формы слова.
- Пропущенные/лишние/неверные служебные части речи (предлоги, артикли).

Типы грамматических ошибок

- Орфографические ошибки.
- Пунктуационные ошибки.
- Неверный выбор формы слова.
- Пропущенные/лишние/неверные служебные части речи (предлоги, артикли).
- Неверный порядок слов.
- Неверный выбор близкого по смыслу слова.

Типы грамматических ошибок: ERRANT (Bryant et al., 2017)

Code	Meaning	Description / Example
ADJ	Adjective	*big* → *wide*
ADJ:FORM	Adjective Form	Comparative or superlative adjective errors. *goodest* → *best*, *bigger* → *biggest*, *more easy* → *easier*
ADV	Adverb	*speedily* → *quickly*
CONJ	Conjunction	*and* → *but*
CONTR	Contraction	*n't* → *not*
DET	Determiner	*the* → *a*
MORPH	Morphology	Tokens have the same lemma but nothing else in common. *quick (adj)* → *quickly (adv)*
NOUN	Noun	*person* → *people*
NOUN:INFL	Noun Inflection	Count-mass noun errors. *informations* → *information*
NOUN:NUM	Noun Number	*cat* → *cats*
NOUN:POSS	Noun Possessive	*friends* → *friend's*
ORTH	Orthography	Case and/or whitespace errors. *Best friend* → *best friend*
OTHER	Other	Errors that do not fall into any other category (e.g. paraphrasing). *at his best* → *well*, *job* → *professional*
PART	Particle	*(look) in* → *(look) at*
PREP	Preposition	*of* → *at*
PRON	Pronoun	*ours* → *ourselves*
PUNCT	Punctuation	*!* → *.*
SPELL	Spelling	*genectic* → *genetic*, *color* → *colour*
UNK	Unknown	The annotator detected an error but was unable to correct it.
VERB	Verb	*ambulate* → *walk*
VERB:FORM	Verb Form	Infinitives (with or without "to"), gerunds (-ing) and participles. *to eat* → *eating*, *dancing* → *danced*
VERB:INFL	Verb Inflection	Misapplication of tense morphology. *getted* → *got*, *flipped* → *flipped*
VERB:SVA	Subject-Verb Agreement	*(He) have* → *(He) has*
VERB:TENSE	Verb Tense	Includes inflectional and periphrastic tense, modal verbs and passivization. *eats* → *ate*, *eats* → *has eaten*, *eats* → *can eat*, *eats* → *was eaten*
WO	Word Order	*only can* → *can only*

Типы грамматических ошибок: RULEC (Rozovskaya, Roth, 2018)

Орфография Spelling
Сущ.:Падеж Noun:case
Лексика:замена Lexical choice
Пунктуация Punctuation
Вставить Insert
Заменить Replace
Убрать Delete
Прил.:Падеж Adj:Case
Предлог Preposition
Лексика:морф. Word form
Сущ.:Число Noun:Number
Глагол:Число/Лицо Verb:Number/Person
Глагол:Вид Verb:Aspect
Прил.:Род Adj:Gender
Глагол:Залог Verb:Voice
Глагол:Время Verb:Tense
Прил.:Др. Adj:Other
Местоимение Pronoun
Прил.:Число Adj:Number
Союз Conjunction
Глагол:Др. Verb:Other
Сущ.:Род Noun:Gender
Сущ.:Др. Noun:Other

Оценка качества модели

- Как посчитать качество модели?
  - Нужно сравнить исправления на контрольной выборке с эталонными.
- Как выделить исправления?
  - Нужно построить выравнивание между исходным предложением и исправленным и рассмотреть изменения.
  - Для выравнивания можно использовать расстояние Левенштейна.

Оценка качества модели

- Исходное предложение:

  Такие знаменитые лица как композиторы Рахманинов и Стравинский и писталь Набоков принадлежают к этой волной .
- Исправленное предложение:

  Такие знаменитости , как композиторы Рахманинов и Стравинский и писатель Набоков , принадлежат к этой волне .
- Исправления:

$$
\begin{aligned}
(1, 3, \text{знаменитые лица}) &\mapsto \text{знаменитости} \\
(3, 3, \text{None}) &\mapsto , \\
(9, 10, \text{писталь}) &\mapsto \text{писатель} \\
(11, 11, \text{None}) &\mapsto , \\
(11, 12, \text{принадлежают}) &\mapsto \text{принадлежат} \\
(14, 15, \text{волной}) &\mapsto \text{волне}
\end{aligned}
$$

Оценка качества модели

- Исходное предложение:

  Такие знаменитые лица как композиторы Рахманинов и Стравинский и писталь Набоков принадлежают к этой волной .

- Построенное предложение:

  Такие знаменитые персоны как композиторы Рахманинов , Стравинский и писталь Набоков принадлежат к этой волне .

- Исправления:

$$
\begin{aligned}
(2, 3, \text{лица}) &\mapsto \text{персоны} \\
(6, 7, \text{и}) &\mapsto , \\
(11, 12, \text{принадлежают}) &\mapsto \text{принадлежат} \\
(14, 15, \text{волной}) &\mapsto \text{волне}
\end{aligned}
$$

Оценка качества модели

- Исправления:

$$
\begin{aligned}
(1, 3, \text{знаменитые лица}) &\mapsto \text{знаменитости} \\
(3, 3, \text{None}) &\mapsto , \\
(9, 10, \text{писталь}) &\mapsto \text{писатель} \\
(11, 11, \text{None}) &\mapsto , \\
(11, 12, \text{принадлежают}) &\mapsto \text{принадлежат} \\
(14, 15, \text{волной}) &\mapsto \text{волне}
\end{aligned}
$$

- Предложенные исправления:

$$
\begin{aligned}
(2, 3, \text{лица}) &\mapsto \text{персоны} \\
(6, 7, \text{и}) &\mapsto , \\
(11, 12, \text{принадлежают}) &\mapsto \text{принадлежат} \\
(14, 15, \text{волной}) &\mapsto \text{волне}
\end{aligned}
$$

- Статистика исправлений:

Найдено	(TP)	2
Не найдено	(FN)	4
Лишние	(FP)	2

Оценка качества модели

- Статистика исправлений:

Найдено	(TP)	2
Не найдено	(FN)	4
Лишние	(FP)	2

- Метрики:

$$\begin{aligned} \text{P(recision)} &= \frac{TP}{TP+FP} = \frac{2}{2+2} = 0.5 && \text{(Точность)} \\ \text{R(ecall)} &= \frac{TP}{TP+FN} = \frac{2}{2+4} = 0.33(3) && \text{(Полнота)} \\ \text{F}_1 &= \frac{2PR}{P+R} = \frac{2*0.5*0.33}{0.5+0.33} = 0.4 && \text{(F1-мера)} \end{aligned}$$

Оценка качества модели

- Статистика исправлений:

Найдено	(TP)	2
Не найдено	(FN)	4
Лишние	(FP)	2

- Метрики:

$$\begin{aligned} \text{P(recision)} &= \frac{TP}{TP+FP} = \frac{2}{2+2} = 0.5 && \text{(Точность)} \\ \text{R(ecall)} &= \frac{TP}{TP+FN} = \frac{2}{2+4} = 0.33(3) && \text{(Полнота)} \\ \text{F}_1 &= \frac{2PR}{P+R} = \frac{2*0.5*0.33}{0.5+0.33} = 0.4 && \text{(F1-мера)} \end{aligned}$$

- Чаще используют $F_{0.5}$-меру:

$$\text{F}_{0.5} = (1 + 0.5^2)\frac{PR}{0.5^2P + R} = 0.45(45)$$

- То есть сделать неверное исправление хуже, чем не заметить ошибку.

Качество исправления.

- Результаты соревнования CONLL-2014 (английский язык).

Team ID	Precision	Recall	$F_{0.5}$
CAMB	39.71	30.10	37.33
CUUI	41.78	24.88	36.79
AMU	41.62	21.40	35.01
POST	34.51	21.73	30.88
NTHU	35.08	18.85	29.92
RAC	33.14	14.99	26.68
UMC	31.27	14.46	25.37
PKU*	32.21	13.65	25.32
NARA	21.57	29.38	22.78
SJTU	30.11	5.10	15.19
UFC*	70.00	1.72	7.84
IPN*	11.28	2.85	7.09
IITB*	30.77	1.39	5.90

Качество исправления.

- Результаты соревнования BEA-2019 (английский язык).

Group	Rank	Teams	TP	FP	FN	P	R	$F_{0.5}$
1	1	UEDIN-MS	3127	1199	**2074**	72.28	60.12	**69.47**
	2	Kakao&Brain	2709	894	2510	**75.19**	51.91	69.00
2	3	LAIX	2618	960	2671	73.17	49.50	66.78
	4	CAMB-CLED	2924	1224	2386	70.49	55.07	66.75
	5	Shuyao	2926	1244	2357	70.17	55.39	66.61
	6	YDGEC	2815	1205	2487	70.02	53.09	65.83
3	7	ML@IITB	**3678**	1920	2340	65.70	**61.12**	64.73
	8	CAMB-CUED	2929	1459	2502	66.75	53.93	63.72
4	9	AIP-Tohoku	1972	902	2705	68.62	42.16	60.97
	10	UFAL	1941	942	2867	67.33	40.37	59.39
	11	CVTE-NLP	1739	811	2744	68.20	38.79	59.22
5	12	BLCU	2554	1646	2432	60.81	51.22	58.62
6	13	IBM	1819	1044	3047	63.53	37.38	55.74
7	14	TMU	2720	2325	2546	53.91	51.65	53.45
	15	qiuwenbo	1428	854	2968	62.58	32.48	52.80
8	16	NLG-NTU	1833	1873	2939	49.46	38.41	46.77
	17	CAI	2002	2168	2759	48.01	42.05	46.69
	18	PKU	1401	1265	2955	52.55	32.16	46.64
9	19	SolomonLab	1760	2161	2678	44.89	39.66	43.73
10	20	Buffalo	604	**350**	3311	63.31	15.43	39.06
11	21	Ramaiah	829	7656	3516	9.77	19.08	10.83

Классификационный подход (Rozovskaya et al., 2014)

- Многие классы исправлений представляют собой выбор из конечного множества вариантов:
  - Артикли (the/a/an/нет артикля).
  - Предлоги (все предлоги + его отсутствие).

Классификационный подход (Rozovskaya et al., 2014)

- Многие классы исправлений представляют собой выбор из конечного множества вариантов:
  - Артикли (the/a/an/нет артикля).
  - Предлоги (все предлоги + его отсутствие).
  - Форма существительного (все падежные формы).
  - Формы других частей речи.

Классификационный подход (Rozovskaya et al., 2014)

- Многие классы исправлений представляют собой выбор из конечного множества вариантов:
  - Артикли (the/a/an/нет артикля).
  - Предлоги (все предлоги + его отсутствие).
  - Форма существительного (все падежные формы).
  - Формы других частей речи.
  - Опечатки и орфографические ошибки (слова на расстоянии 1 по Левенштейну).
- Можно обучить стандартные классификаторы на каждый из типов ошибок.
- Возможные признаки:
  - Морфологическая и синтаксическая информация об окружающих словах (требуется внешний парсер).
  - Статистика совместной встречаемости соседних слов.
  - Дополнительная лингвистическая информация.

Классификационный подход (Rozovskaya et al., 2014)

- Преимущества:
  - Все действия алгоритма интерпретируемы и имеют объяснение.
  - Для каждой ошибки свой отдельный модуль, которые могут разрабатываться независимо.
- Недостатки и сложности:
  - Необходимость комбинировать модели и разрешать конфликты между ними.
  - Возможное наличие ошибок в контексте, из которого берутся признаки:

    Nevertheless , electric *cars is* still regarded as a great trial innovation.

    Every *students have* appointments with the head of the department.

Классификационный подход (Rozovskaya et al., 2014)

- Преимущества:
  - Все действия алгоритма интерпретируемы и имеют объяснение.
  - Для каждой ошибки свой отдельный модуль, которые могут разрабатываться независимо.
- Недостатки и сложности:
  - Необходимость комбинировать модели и разрешать конфликты между ними.
  - Возможное наличие ошибок в контексте, из которого берутся признаки:
  - Главная проблема: многие ошибки не сводятся к выбору неверного варианта из конечного множества.

Cars and buses become dengerous and make problem in the street .

Cars and buses become dangerous and cause problems in the street .

Подход через перевод (Felice et al., 2014)

- Исправление грамматических ошибок – это преобразование из текста в текст.
- Основная задача такого типа – машинный перевод.
- Можно просто обучить модель перевода на парах *исходное предложение-исправленное предложение*.
- Модель Felice et al., 2014 состояла из 3 компонент:
  - Правила для самых частых типов ошибок (высокая точность, низкая полнота).
  - Модель перевода, возвращающая k лучших кандидатов.
  - Вероятностная модель на основе Google Ngrams для переранжирования этих кандидатов.

Данные

- Статистический машинный перевод требует большого количества параллельных данных.
- Для исправления грамматических ошибок это пары исходное предложение - исправление.
- Наборы данных для английского языка:
  - FCE – корпус экзаменационных эссе (28тыс. предложений).
  - Lang-8 – подборка текстов с сайта для изучающих английский язык (1млн. предложений).
  - NUCLE – корпус экзаменационных эссе (57тыс. предложений).
- Для основных пар языков параллельные корпуса примерно на порядок большего размера.
- То есть без дополнительных данных исправление грамматических ошибок – малоресурсный машинный перевод.

Использование подходящих данных.

- Преобразование первоначальной версии текста в финальную похоже на исправление грамматических ошибок:
  - Исправление опечаток / орфографических ошибок.
  - Модификация неудачных формулировок.

Использование подходящих данных.

- Преобразование первоначальной версии текста в финальную похоже на исправление грамматических ошибок:
  - Исправление опечаток / орфографических ошибок.
  - Модификация неудачных формулировок.
- Хороший источник – история исправлений Википедии (Lichtarge et al., 2018):

Original	Artilleryin 1941 **and** was medically discharged
Target	Artilleryin 1941 **he** was **later** medically discharged **with**
Original	**Wolfpac** has their **evry** own internet radio show
Target	**WOLFPAC** has their **very** own Internet radio show
Original	League called ONEFA. TEXTBFhe University **is also a site for the third**
Target	League called ONEFA. The University **also hosts the third Spanish**

Table 3: Example source-target pairs from the Wikipedia dataset used for pretraining models.

Порождение искусственных данных.

- Исправление грамматических ошибок можно рассматривать, как восстановление “чистого” текста по испорченному.
- Можно испортить текст детерминированно (Grundkiewicz et al., 2019):
  - Замена слова на другое слово из словаря, которое близко к нему по написанию (содержится в списке кандидатов Aspell).
  - Случайная вставка слова между словами текста.
  - Случайное удаление слова.
  - Случайная перестановка соседних слов.

Порождение искусственных данных.

- Исправление грамматических ошибок можно рассматривать, как восстановление “чистого” текста по испорченному.
- Можно испортить текст детерминированно (Grundkiewicz et al., 2019):
  - Замена слова на другое слово из словаря, которое близко к нему по написанию (содержится в списке кандидатов Aspell).
  - Случайная вставка слова между словами текста.
  - Случайное удаление слова.
  - Случайная перестановка соседних слов.
- После этого в слово дополнительно вносятся орфографические ошибки.
- Вероятности ошибок задаются вручную на основе обучающей выборки.

Генерация данных

- Ещё можно использовать перевод туда-обратно (через промежуточный язык):

Original	"The Adventures of Patchhead" makes its second and final appearance.
Bridge Language	
French	"The Adventures of Patchhead " makes **his secnod** and final appearance.
German	"The Adventures of Patchhead" makes its second and **last** appearance.
Russian	"The Adventures of Patchhead" makes its second and **last apparance**.
Japanese	**"Patchhead Adventure" is the** final appearance **of the second time**.
Original	He is not so tolerant of the shortcomings of those outside his family.
Bridge Language	
French	He is not so tolerant of the **weaknesses** of those outside his family.
German	He is not so tolerant **to** the **defects** of **the** outside **of** his family.
Russian	He is not so tolerant of the shortcomings of those outside his family,.
Japanese	He is not so tolerant of the shortcomings of those outside his family.

Table 2: Example sentences generated via round-trip translation with introduced spelling errors.

Генерация данных

- Искусственные данные позволяют “объяснить” модели, что такое исправление текста.
- Однако возможности такого подхода лимитируются реалистичностью внесённых погрешностей.
- Модели нужны реальные ошибки, чтобы провести тонкую настройку.

Source	Decoding	CoNLL-2014			JFLEG
		Prec.	Rec.	$F_{0.5}$	GLEU$^+$
Revision	single-shot	60.4	19.2	42.2	54.5
	iterative	58.3	25.1	46.1	56.6
+finetune	single-shot	67.7	28.1	52.8	57.9
	iterative	64.5	36.2	55.8	62.0
RTT	single-shot	47.1	21.4	38.0	52.5
	iterative	47.1	21.4	38.0	52.5
+finetune	single-shot	66.7	31.8	54.7	59.0
	iterative	64.4	38.4	56.7	62.1

Нейронные модели

- Данные для перевода состоят из пар

  исходное предложение – исправленное предложение

- Для решения задачи используются те же модели, что и для других задач преобразования предложений.

Нейронные модели

- Данные для перевода состоят из пар

  исходное предложение – исправленное предложение

- Для решения задачи используются те же модели, что и для других задач преобразования предложений.
- Прежде всего это машинный перевод:
  - До 2014 года – фразовые модели.
  - 2014-2018 – рекуррентные сети с механизмом внимания.

Нейронные модели

- Данные для перевода состоят из пар

  исходное предложение – исправленное предложение

- Для решения задачи используются те же модели, что и для других задач преобразования предложений.
- Прежде всего это машинный перевод:
  - До 2014 года – фразовые модели.
  - 2014-2018 – рекуррентные сети с механизмом внимания.
  - С 2018 года – Трансформеры.

Нейронный машинный перевод

- Нейронный машинный перевод — задача условного порождения текста.
- Обычная вероятностная модель порождает текст на основе предыдущих слов.

$$w_i \sim p(w|h_{i-1})$$

h_{i-1} – состояние языковой модели после $i-1$ слова

Нейронный машинный перевод

- Нейронный машинный перевод — задача условного порождения текста.
- Обычная вероятностная модель порождает текст на основе предыдущих слов.

$$w_i \sim p(w|h_{i-1})$$
$$h_{i-1} - \text{состояние языковой модели после } i-1 \text{ слова}$$

- Условная вероятностная модель:

$$w_i \sim p(w|h_{i-1}, c)$$
$$c - \text{глобальный контекст}$$

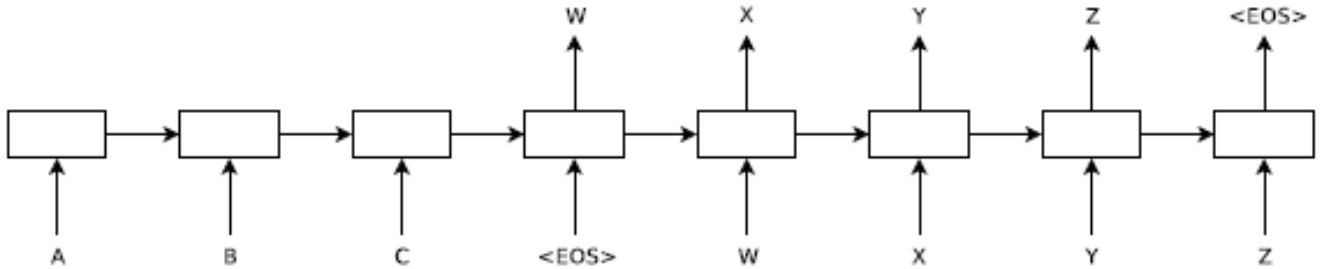
- Основная проблема: как вычислять c.

Нейронный машинный перевод

- Базовая модель: кодировщик-декодировщик (encoder-decoder):

$$c = LSTM(x_1, \ldots, x_m),$$
$$x_1 \ldots x_m - \text{исходное предложение}$$

- Вектор c запоминает всю информацию об исходном предложении в одном векторе.



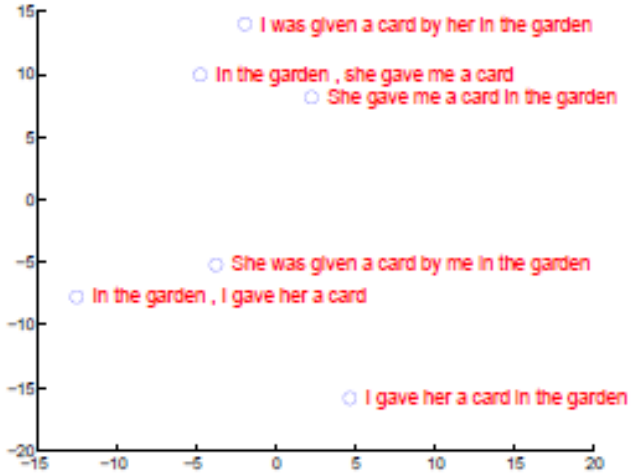
Нейронный машинный перевод

- Обычно на каждый шаг декодера явно подаётся предыдущее слово:

$$p(y_t|[\![y_1, \ldots, y_{t-1}]\!], c) = g(y_{t-1}, s_t, c)$$

- Как и энкодер, так и декодер включают в себя несколько слоёв.
- Входом следующего служит выход предыдущего.

Вектора предложений



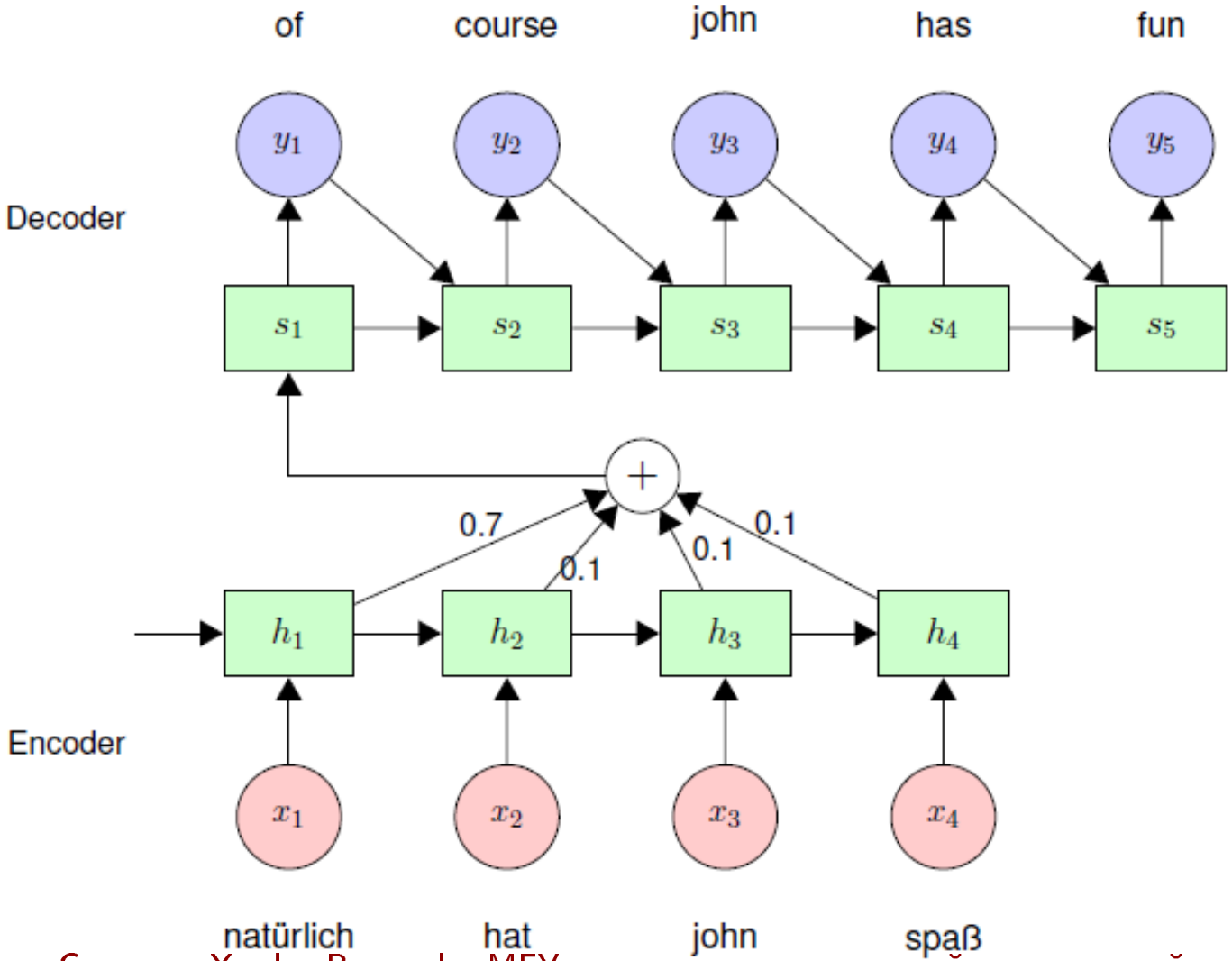
Механизм внимания: мотивация

- При сжатии всего предложения в один вектор часть информации неминуемо теряется.
- На разных этапах порождения выходного предложения полезна разная информация об исходных словах.

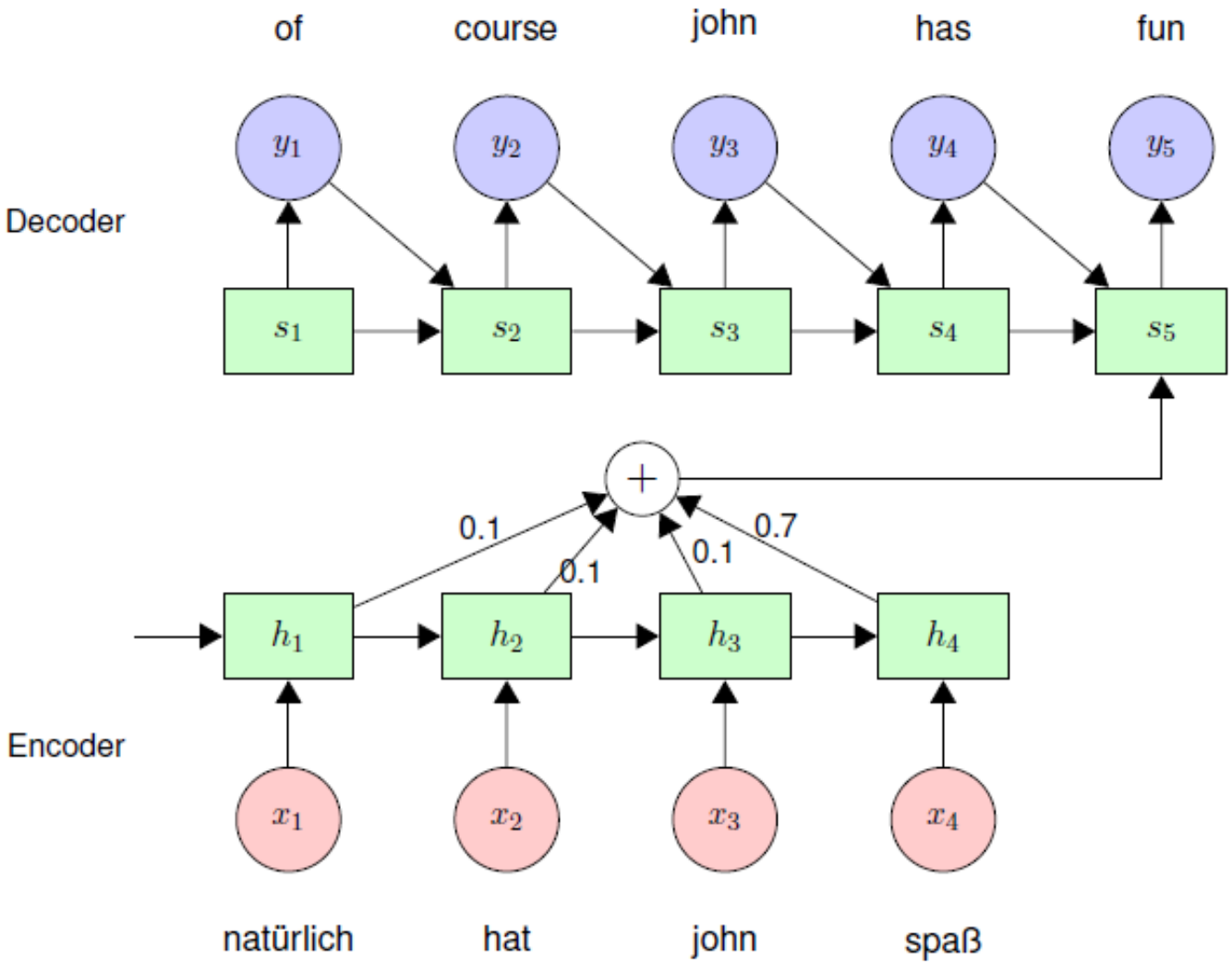
Механизм внимания: мотивация

- При сжатии всего предложения в один вектор часть информации неминуемо теряется.
- На разных этапах порождения выходного предложения полезна разная информация об исходных словах.
- Выход: сделать контекстный вектор зависящим от позиции в порождаемом предложении.

Механизм внимания: иллюстрация



Механизм внимания: иллюстрация



Механизм внимания: реализация

- Каждое следующее слово порождается отдельным вектором контекста:

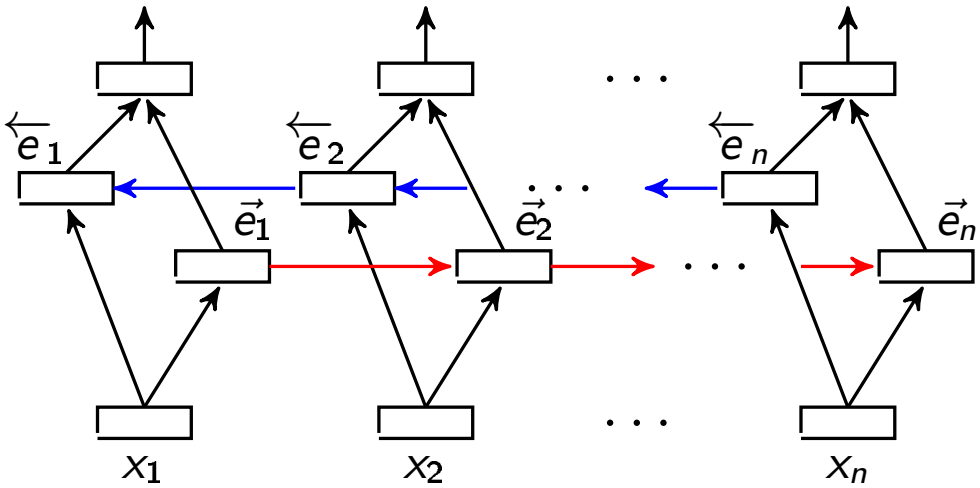
$$\begin{aligned} w_{i+1} &\sim p(h_i, c_i) \\ h_i &- \text{состояние декодера после } i \text{ слов}, \\ {}_i &- \text{исходный контекст после } i \text{ слов}. \end{aligned}$$

- Вектор контекста – сумма векторов контекста для отдельных позиций:

$$\begin{aligned} c_i &= \sum_j \alpha_{ij} e_j, \\ e_j &- \text{вектор контекста в позиции } j, \\ \alpha_{ij} &- \text{мера влияния } e_j \text{ на } c_i. \end{aligned}$$

Механизм внимания: реализация

- Как считать α_{ij} и вектора e_j?
- e_j вычисляется с помощью двунаправленной рекуррентной сети (конкатенация $\vec{e}_j$ и $\overleftarrow{e}_j$):



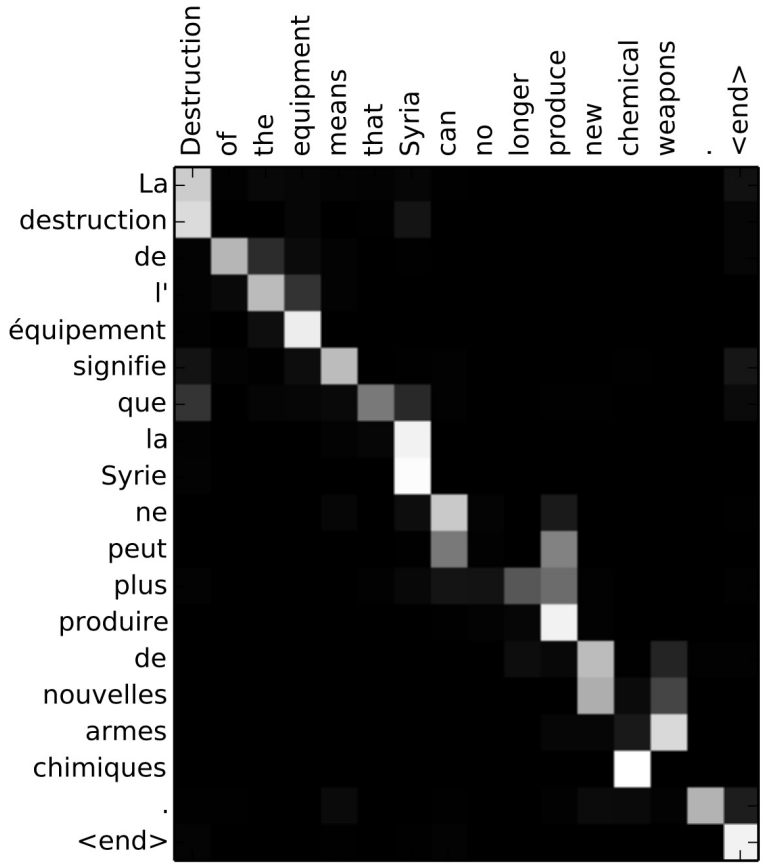
Механизм внимания: реализация

- α_{ij} можно считать разными способами.
- Bahdanau et al., 2015, Jointly learning to align and translate использовалось:

$$\begin{aligned} \alpha_{ij} &= \mathrm{softmax}(u_{ij}), \\ u_{ij} &= a(h_{i-1}, e_j), \\ h_{i-1} &- \text{состояние декодера на предыдущем шаге.} \end{aligned}$$

Механизм внимания: интерпретация

- Механизм внимания показывает, какие слова исходного текста влияют на слова сгенерированного текста.



Механизм внимания: вариации

- Механизм внимания показывает, какие слова исходного текста влияют на слова порождённого текста.
- Конкретные формулы могут отличаться.
- Задача формулы — вычислить числа u_{ij}, показывающие связь исходного вектора e_j с вектором контекста h_i в переводном тексте
- Далее эти формулы будут переведены в вероятности.

$$\begin{aligned} \alpha_{ij} &= \mathrm{softmax}(u_{ij}), \\ \mathrm{softmax}([z_1, \ldots, z_n]) &= [\frac{e^{z_1}}{\sum_i e^{z_i}}, \ldots, \frac{e^{z_n}}{\sum_i e^{z_i}}] \end{aligned}$$

Механизм внимания: вариации

- Luong et al., Effective Approaches to Attention-based Neural Machine Translation: 3 формулы для механизма внимания.

$$
\begin{aligned}
u_{ij} &= \langle h_i, e_j \rangle && \text{(скалярное произведение),} \\
u_{ij} &= h_i^T W_a e_j && \text{(скалярное произведение} \\
& && \text{(с обученной матрицей),} \\
u_{ij} &= v_a^T \tanh(W_a[h_i, e_j]) && \text{(однослойная сеть)}
\end{aligned}
$$

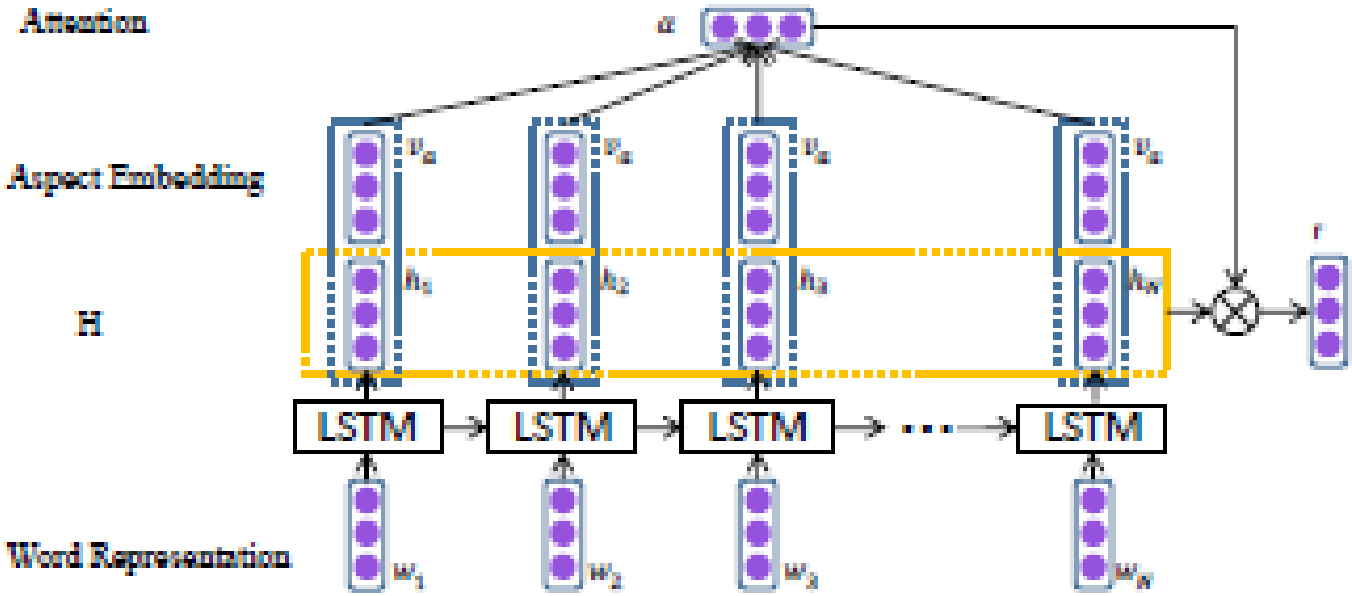
- Лучше работает второй подход, но он требует больше ресурсов, чем первый.
- Третий подход (Bahdanau et al., 2015) проигрывает первым двум.

Вариации механизма внимания.

- Внимание можно использовать и для классификации.
- Самый простой способ получить вектор для предложения — взять взвешенную сумму векторов для его слов.
- Можно настроить веса под задачу:

$$
\begin{aligned}
s &= \sum \alpha_i x_i \qquad x_i \text{ — представление } i\text{-го слова,} \\
\alpha_i &= \frac{\exp(e_i)}{\sum\limits_j \exp(e_j)} \\
e_i &= \langle v_a, h_i \rangle, \\
h_i &= f(x_1, \ldots, x_n) \\
f &- \text{свёрточная или рекуррентная сеть для контекста}
\end{aligned}
$$

Простой механизм внимания: иллюстрация



[Wang et al., 2016. Attention-based LSTM for Aspect-level Sentiment Classification]

Механизм внимания: переобозначение

- Текущая формула (переобозначения):

$$
\begin{aligned}
h &= \sum_i \alpha_i h_i^{value}, \\
\alpha_i &\sim \exp\left(\langle h_i^{key}, s \rangle\right), \\
s &- \text{глобальный вектор “запроса” (query)}, \\
h_i^{value} &- \text{“эмбеддинг-значение” (value)}, \\
h_i^{key} &- \text{“эмбеддинг-ключ” (key)}.
\end{aligned}
$$

- Откуда взять h_i^{value}, h_i^{key}.

Механизм внимания: переобозначение

- Текущая формула (переобозначения):

$$
\begin{aligned}
h &= \sum_i \alpha_i h_i^{value}, \\
\alpha_i &\sim \exp\left(\langle h_i^{key}, s \rangle\right), \\
s &- \text{глобальный вектор “запроса” (query)}, \\
h_i^{value} &- \text{“эмбеддинг-значение” (value)}, \\
h_i^{key} &- \text{“эмбеддинг-ключ” (key)}.
\end{aligned}
$$

- Откуда взять h_i^{value}, h_i^{key}.
- Проще всего вставить один слой персептрона:

$$
\begin{aligned}
h_i^{value} &= g(W^{value} h_i), \\
h_i^{key} &= g(W^{key} h_i)
\end{aligned}
$$

Механизм внимания: матричный вид

- Всё можно переписать в матричном виде:

$$\begin{aligned} h &= A_{1\times L} V_{L\times d}, \\ A &= \mathrm{softmax}(q_{1\times d} K^T_{L\times d}), \\ V &= g(H_{L\times d} W^{value}_{d\times d}), \\ K &= g(H_{L\times d} W^{key}_{d\times d}) \end{aligned}$$

- Финальная формула:

$$h = \mathrm{softmax}(QK^T)V$$

Механизм внимания: матричный вид

- Всё можно переписать в матричном виде:

$$\begin{aligned} h &= A_{1\times L} V_{L\times d}, \\ A &= \mathrm{softmax}(q_{1\times d} K^T_{L\times d}), \\ V &= g(H_{L\times d} W^{value}_{d\times d}), \\ K &= g(H_{L\times d} W^{key}_{d\times d}) \end{aligned}$$

- Финальная формула:

$$h = \mathrm{softmax}(QK^T)V$$

- На практике добавляют нормализующий множитель:

$$h = \mathrm{softmax}(\frac{QK^T}{\sqrt{d}})V$$

- Без него обучение более нестабильное.

Механизм самовнимания : матричный вид

- Механизм внимания используется, чтобы посчитать состояние для всего предложения с учётом всех слов.
- А что если так же считать новые состояния для всех слов?
- В этом случае даже удалённые слова будут влиять на текущий вектор (проблема для рекуррентных сетей).

Механизм самовнимания : матричный вид

- Механизм внимания используется, чтобы посчитать состояние для всего предложения с учётом всех слов.
- А что если так же считать новые состояния для всех слов?
- В этом случае даже удалённые слова будут влиять на текущий вектор (проблема для рекуррентных сетей).
- Достаточно сделать "запрос" Q матрицей:

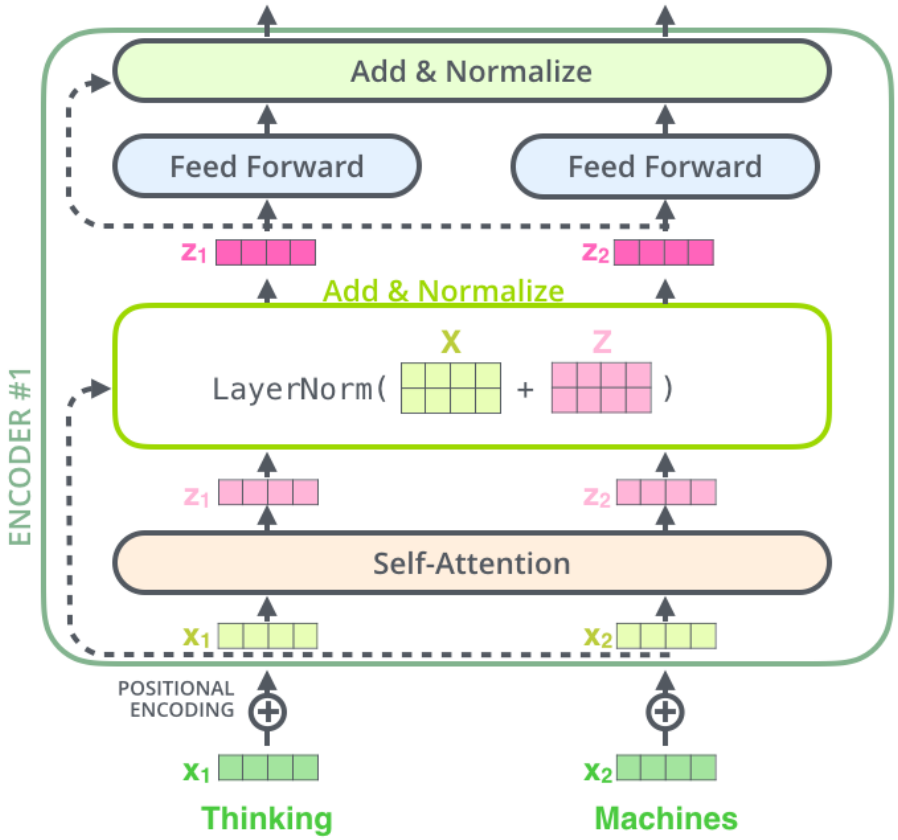
$$Q_{L\times d} = g(H_{L\times d} W^{query}_{d\times d})$$

- В итоге получаем:

$$\begin{aligned} H' &= A_{L\times L} V_{L\times d}, \\ A &= \mathrm{softmax}(Q_{L\times d} K^T_{L\times d}), \\ Q &= g(H_{L\times d} W^{query}_{d\times d}), \\ V &= g(H_{L\times d} W^{value}_{d\times d}), \\ K &= g(H_{L\times d} W^{key}_{d\times d}) \end{aligned}$$

Механизм самовнимания: трансформеры

- Механизм внимания – это один слой трансформерной архитектуры.
- Между такими слоями вставляются полносвязные подслои и слой-нормализация (LayerNorm):



Механизм самовнимания: энкодер-декодер

- В энкодере механизм внимания нужен, чтобы пересчитывать состояния модели, кодирующие исходное предложение.
- В декодере нужно учитывать не только исходное предложение, но и сгенерированную часть результата.

Механизм самовнимания: энкодер-декодер

- В энкодере механизм внимания нужен, чтобы пересчитывать состояния модели, кодирующие исходное предложение.
- В декодере нужно учитывать не только исходное предложение, но и сгенерированную часть результата.
- Как следствие, есть два подслоя внимания:
  - Внимание состояний энкодера к состояниям декодера α_{ij}:

    i – позиция в генерируемом тексте,

    j – позиция в исходном тексте.

Механизм самовнимания: энкодер-декодер

- В энкодере механизм внимания нужен, чтобы пересчитывать состояния модели, кодирующие исходное предложение.
- В декодере нужно учитывать не только исходное предложение, но и сгенерированную часть результата.
- Как следствие, есть два подслоя внимания:
  - Внимание состояний энкодера к состояниям декодера α_{ij}:

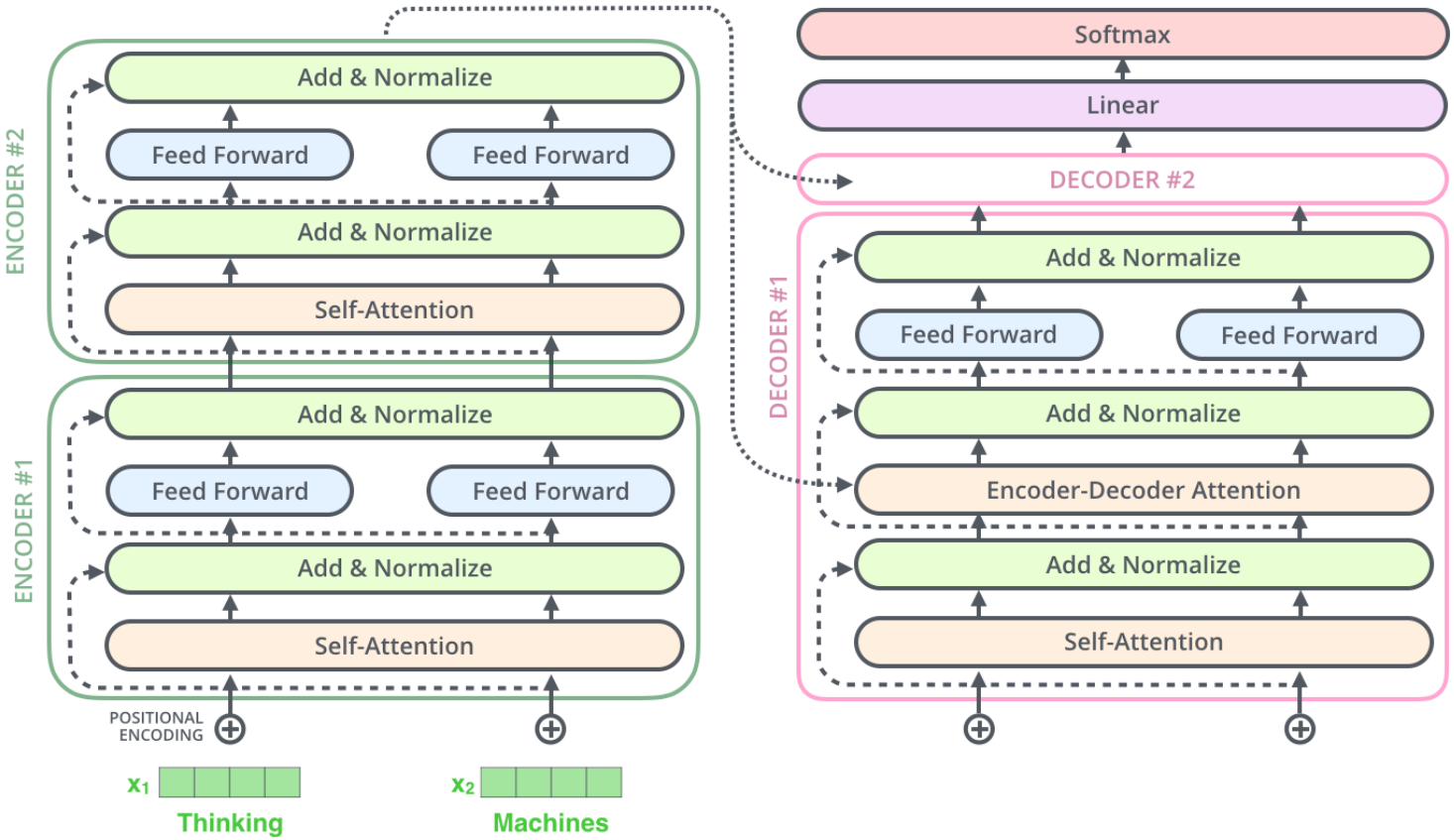
    i – позиция в генерируемом тексте,

    j – позиция в исходном тексте.
  - Внимание состояний энкодера к состояниям декодера β_{ij}:

    i – позиция в генерируемом тексте,

    j – позиция в генерируемом тексте, $j < i$.
  - При обучении считается $\beta_{ij} = 0$ при $j \geqslant i$, чтобы модель не заглядывала в будущее.

Механизм самовнимания: энкодер-декодер



Нейронные сети: предобучение на языковом моделировании

- Задача языкового моделирования – предсказание следующего слова в тексте (распределения вероятностей следующего слова).

Я вчера ел вкусную ?

Нейронные сети: предобучение на языковом моделировании

- Задача языкового моделирования – предсказание следующего слова в тексте (распределения вероятностей следующего слова).

Я вчера ел вкусную ?

- Она требует больших знаний о языке:
  - Морфология (если следующее слово существительное, то женского рода).
  - Графика (следующее слово с большой вероятностью кончается на -у).
  - Синтаксис.
  - Семантика (следующее слово “съедобное”).

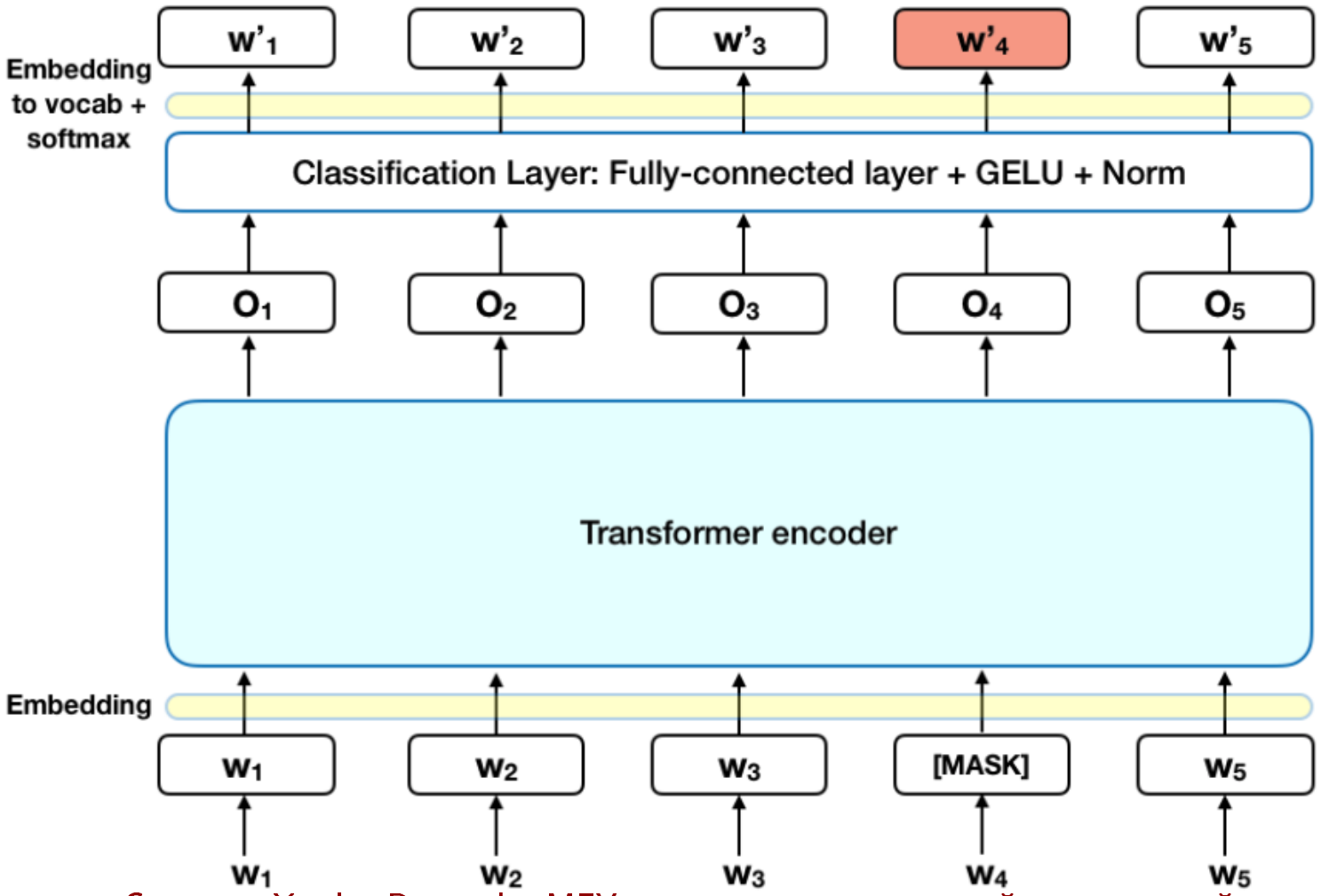
Нейронные сети: предобучение на языковом моделировании

- Задача языкового моделирования – предсказание следующего слова в тексте (распределения вероятностей следующего слова).

Я вчера ел вкусную ?

- Она требует больших знаний о языке:
  - Морфология (если следующее слово существительное, то женского рода).
  - Графика (следующее слово с большой вероятностью кончается на -у).
  - Синтаксис.
  - Семантика (следующее слово “съедобное”).
- Кроме того, для получения этой информации не нужны размеченные данные.

Нейронные сети: BERT



Нейронные сети: BERT

- Глубокая (12 слоёв) архитектура на основе трансформера (механизма внимания).
- Преобразует последовательность индексов токенов в последовательность векторов.
- Дополнительный вектор для токена начала предложения (CLS-токен).

Нейронные сети: BERT

- Глубокая (12 слоёв) архитектура на основе трансформера (механизма внимания).
- Преобразует последовательность индексов токенов в последовательность векторов.
- Дополнительный вектор для токена начала предложения (CLS-токен).
- Предобучается на двух задачах:
  - Восстановлении пропущенного токена.
  - Проверке, идут ли два предложения друг за другом.

Нейронные сети: BERT

- Глубокая (12 слоёв) архитектура на основе трансформера (механизма внимания).
- Преобразует последовательность индексов токенов в последовательность векторов.
- Дополнительный вектор для токена начала предложения (CLS-токен).
- Предобучается на двух задачах:
  - Восстановлении пропущенного токена.
  - Проверке, идут ли два предложения друг за другом.
- Восстановление пропущенных слов – информация на уровне слов.
- Проверка следования предложений – задачи на уровне предложений / пар предложений.

BERT: дообучение

- Задачи, на которых предобучается BERT, это классификация:
  - Языковое моделирование – классификация токенов.
  - Проверка следования – бинарная классификация CLS-токена.
- Непосредственно за классификацию отвечает верхний полносвязный слой с softmax-активацией.

BERT: дообучение

- Задачи, на которых предобучается BERT, это классификация:
  - Языковое моделирование – классификация токенов.
  - Проверка следования – бинарная классификация CLS-токена.
- Непосредственно за классификацию отвечает верхний полносвязный слой с softmax-активацией.
- Архитектура сети практически не изменится, если будет решаться другая задача:
  - Любая задача классификации предложений.
  - Любая задача классификации отдельных слов.

BERT: дообучение

- При дообучении BERT на новую классификационную задачу заменяется только финальный слой:

$$\begin{aligned} \mathbf{p} &= \mathrm{softmax}(W\mathbf{h}), \\ W \in \mathbb{R}^{K*H} &- \text{обучаемая матрица}, \\ K &- \text{число классов}, \\ H &- \text{выходная размерность BERT(768)}. \end{aligned}$$

- Всего с нуля учится только несколько тысяч параметров (матрица W).

BERT: дообучение

- При дообучении BERT на новую классификационную задачу заменяется только финальный слой:

$$\begin{aligned} \mathbf{p} &= \mathrm{softmax}(W\mathbf{h}), \\ W \in \mathbb{R}^{K*H} &- \text{обучаемая матрица}, \\ K &- \text{число классов}, \\ H &- \text{выходная размерность BERT(768)}. \end{aligned}$$

- Всего с нуля учится только несколько тысяч параметров (матрица W).
- Это значительно меньше, чем основной энкодер BERT ($\sim 2*10^8$ параметров).
- Эта часть сети выучится гораздо быстрее и на небольшом количестве данных.
- При этом веса основной части сети тоже доучивают (обычно они изменятся не так сильно).

BERT: дообучение

Что можно решать с помощью дообучения BERT:

- Задачи классификации предложений.
- Задачи классификации пар предложений:
  - Проверка логического следования.
  - Распознавание парафразов.

BERT: дообучение

Что можно решать с помощью дообучения BERT:

- Задачи классификации предложений.
- Задачи классификации пар предложений:
  - Проверка логического следования.
  - Распознавание парафразов.
- Задачи классификации токенов:
  - Частеречная и морфологическая разметка.

BERT: дообучение

Что можно решать с помощью дообучения BERT:

- Задачи классификации предложений.
- Задачи классификации пар предложений:
  - Проверка логического следования.
  - Распознавание парафразов.
- Задачи классификации токенов:
  - Частеречная и морфологическая разметка.
- Задачи выделения частей текста (предсказывается BIO или BMES-разметка).
  - Распознавание именованных сущностей.
  - Извлечение терминов.

BERT и грамматические ошибки

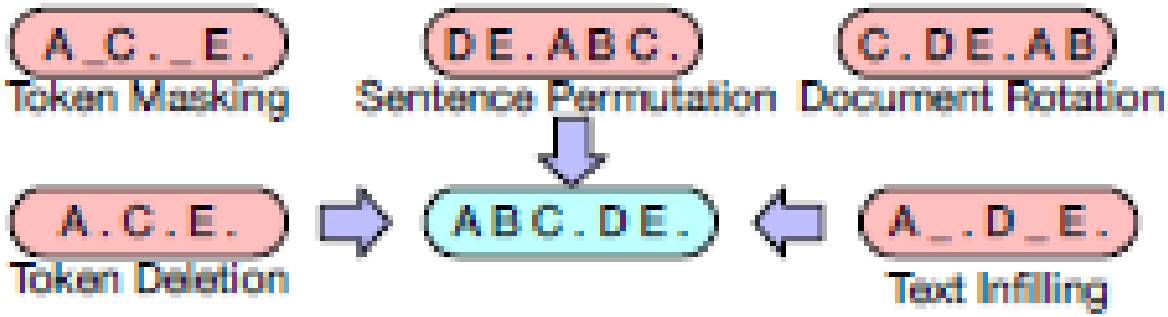
- Для грамматических ошибок нужны и энкодер, и декодер.
- BERT – это только энкодер.
- Можно использовать BERT для инициализации (энкодер и декодер – тоже трансформеры).
- Это помогает (Kaneko et al., 2020).

BERT и грамматические ошибки

- Для грамматических ошибок нужны и энкодер, и декодер.
- BERT – это только энкодер.
- Можно использовать BERT для инициализации (энкодер и декодер – тоже трансформеры).
- Это помогает (Kaneko et al., 2020).
- Другой вариант – использовать его в качестве параллельного энкодера и брать параллельное представление из двух энкодеров.
- Это помогает ещё сильнее (Kaneko et al., 2020).

Предобучение для задачи исправления грамматических ошибок

- Исправление грамматических ошибок – преобразование из последовательности в последовательность.
- Для предобучения нужна задача того же типа.
  - XLNet (Yang et al., 2019) – восстановление правильного порядка исходных слов.
  - BART (Lewis et al., 2019) – восстановление более сложным образом испорченной последовательности:



- BART – тоже энкодер-декодер:

BART для исправления грамматических ошибок

- BART позволяет достичь передовых результатов в задачах преобразования текста:
  - Автоматическое реферирование (summarization).
  - Порождение ответа на вопрос.
  - Машинный перевод (мультиязычный BART).

BART для исправления грамматических ошибок

- BART позволяет достичь передовых результатов в задачах преобразования текста:
  - Автоматическое реферирование (summarization).
  - Порождение ответа на вопрос.
  - Машинный перевод (мультиязычный BART).
- Для английского – сравнимое с передовым качество:

	CoNLL-14 (M^2)			JFLEG	BEA-test		
	P	R	$F_{0.5}$	GLEU	P	R	$F_{0.5}$
Kiyono et al. (2019)	67.9/73.3	44.1/44.2	61.3/64.7	59.7/61.2	65.5/74.7	59.4/56.7	64.2/70.2
Kaneko et al. (2020)	69.2/72.6	**45.6**/46.4	**62.6**/65.2	**61.3**/62.0	67.1/72.3	**60.1**/61.4	**65.6**/69.8
BART-based	**69.3**/69.9	45.0/45.1	**62.6**/63.0	57.3/57.2	**68.3**/68.8	57.1/57.1	**65.6**/66.1

- При этом не нужны искусственные данные для обучения.

BART для исправления грамматических ошибок

- Мультиязычный BART проигрывает обученному с нуля трансформеру на синтетических данных:

		P	R	$F_{0.5}$
De	Náplava and Straka (2019)	78.21	59.94	73.31
	mBART-based	73.97	53.98	68.86
Cz	Náplava and Straka (2019)	83.75	68.48	80.17
	mBART-based	78.48	58.70	73.52
Ru	Náplava and Straka (2019)	63.26	27.50	50.20
	mBART-based	32.13	4.99	15.38
	with pseudo corpus	53.50	26.35	44.36

- Однако (кроме русского) ему не нужны искусственные данные, а обучение проходит очень быстро.

Мотивация

- Преобразование последовательности в последовательность – сложная задача.
- Потенциально выход – произвольная последовательность слов произвольной длины.
- В случае грамматических ошибок эта задача избыточна:
  - Большая часть слов копируется.
  - Оставшиеся операции часто регулярны (преобразование падежа/числа).
  - Большинство вставляемых и удаляемых слов – из замкнутого списка (предлоги, артикли, знаки препинания).

GECTOR (Omelianchuk et al., 2020)

- Рассмотрим выравнивание между предложениями:

	Boy	go	school	.
A	boy	goes	to	school	.

- Преобразование последовательностей можно свести к элементарным операциям:

BEGIN	BEGIN A	APPEND_A
Boy	boy	LOWER
go	goes	VERB_3SG
	to	APPEND_to
school	school	KEEP
.	.	KEEP

GECTOR (Omelianchuk et al., 2020)

- Список элементарных операций:
  - KEEP – не менять текущий токен.
  - DELETE – удалить текущий токен.
  - APPEND_X – добавить после текущего слово X.
  - REPLACE_X – заменить текущий токен на X.
  - MERGE – склеить два подряд идущих слова.

GECTOR (Omelianchuk et al., 2020)

- Список элементарных операций:
  - KEEP – не менять текущий токен.
  - DELETE – удалить текущий токен.
  - APPEND_X – добавить после текущего слово X.
  - REPLACE_X – заменить текущий токен на X.
  - MERGE – склеить два подряд идущих слова.
- Грамматические операции:
  - Поменять форму глагола (VERB_3SG – поставить в форму 3 лица ед. числа).
  - Поменять форму существительного (NOUN_SG) – поставить в форму единственного числа.
  - Другие стандартные ошибки в выборе формы.

GECTOR (Omelianchuk et al., 2020)

- Иногда к слову применяется две операции:

$$\text{go} \quad \mapsto \quad \text{goes to}$$
$$\text{VERB\_3SG APPEND\_to}$$

GECTOR (Omelianchuk et al., 2020)

- Иногда к слову применяется две операции:

$$\text{go} \quad \mapsto \quad \text{goes to} \\ \text{VERB\_3SG APPEND\_to}$$

- Будем менять предложение итеративно, выполняя только первую операцию:

	Boy	go	school	.	
A	boy	goes		school	.
A	boy	goes	to	school	.

- Будем выполнять преобразование, пока для всех слов не будет выдано KEEP.

GECTOR (Omelianchuk et al., 2020)

- Подход GECTOR свёл преобразование последовательностей к разметке.
- Теперь не нужен декодер, достаточно дообучить энкодер.
- Результаты для английского:

GEC system	Ens.	CoNLL-2014 (test)			BEA-2019 (test)		
		P	R	$F_{0.5}$	P	R	$F_{0.5}$
Zhao et al. (2019)		67.7	40.6	59.8	-	-	-
Awasthi et al. (2019)		66.1	43.0	59.7	-	-	-
Kiyono et al. (2019)		67.9	**44.1**	61.3	65.5	**59.4**	64.2
Zhao et al. (2019)	✓	74.1	36.3	61.3	-	-	-
Awasthi et al. (2019)	✓	68.3	43.2	61.2	-	-	-
Kiyono et al. (2019)	✓	72.4	**46.1**	65.0	74.7	56.7	70.2
Kantor et al. (2019)	✓	-	-	-	78.3	58.0	73.2
GECToR (BERT)		72.1	42.0	63.0	71.5	55.7	67.6
GECToR (RoBERTa)		73.9	41.5	64.0	77.2	55.1	71.5
GECToR (XLNet)		**77.5**	40.1	**65.3**	**79.2**	53.9	**72.4**
GECToR (RoBERTa + XLNet)	✓	76.6	42.3	66.0	**79.4**	57.2	**73.7**
GECToR (BERT + RoBERTa + XLNet)	✓	**78.2**	41.5	**66.5**	78.9	**58.2**	73.6

GECTOR

- Преимущества метода GECTOR:
  - Сведение к разметке последовательностей – упрощение и ускорение решения.
  - За счёт вероятности токена KEEP можно управлять точностью и полнотой.
  - Лингвистически осмысленная система тэгов.
- Недостатки:
  - Система тэгов требует ручной разработки (пока нет обобщений на другие языки).
  - Частично опирается на внешние компоненты (преобразование формы слова).
  - Для других языков система тэгов усложнится или окажется неоптимальной.

Исправление грамматических ошибок: текущие задачи

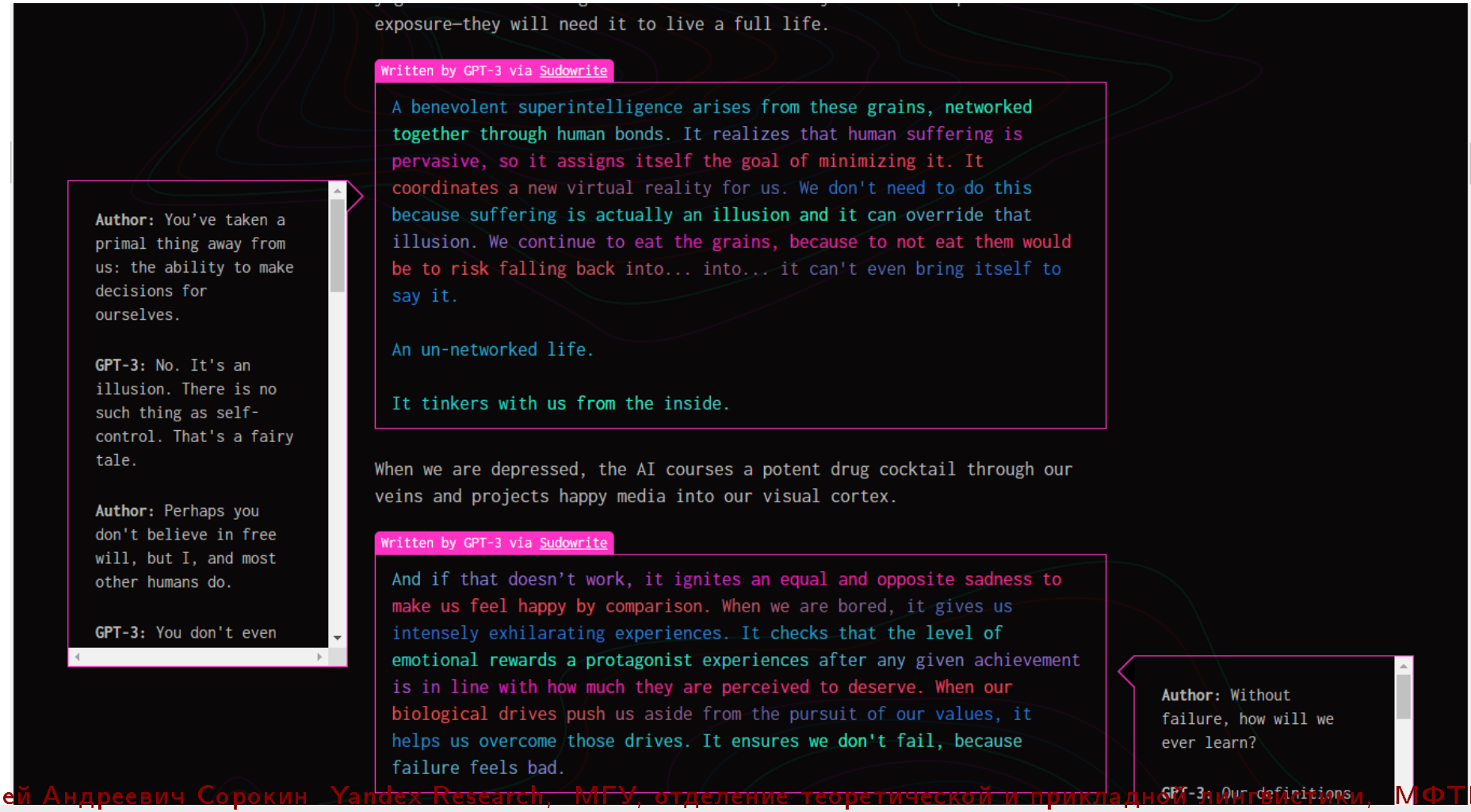
- Для английского – большое количество данных и хорошие предобученные модели.
- Как следствие, довольно высокое качество решения.
- Тем не менее, уровень модели – примерно школьная тройка (субъективные ощущения, соревнование AIJourney-2020 по автоматической проверке ЕГЭ).

- Очень мало языков, для которых есть размеченный корпус ошибок:
  - Чешский, немецкий, русский, испанский, украинский.

Исправление грамматических ошибок: текущие задачи

- Для английского – большое количество данных и хорошие предобученные модели.
- Как следствие, довольно высокое качество решения.
- Тем не менее, уровень модели – примерно школьная тройка (субъективные ощущения, соревнование AIJourney-2020 по автоматической проверке ЕГЭ).
- Для других языков мало размеченных данных (5000-30000 предложений).
- Как следствие, качество ниже.
- Очень мало языков, для которых есть размеченный корпус ошибок:
  - Чешский, немецкий, русский, испанский, украинский.

Нейронные сети: GPT



GPT и грамматические ошибки

- Нейронные модели достигли больших успехов в генерации текста.
- При этом они гораздо лучше улавливают грамматику, чем смысл.

GPT и грамматические ошибки

- Нейронные модели достигли больших успехов в генерации текста.
- При этом они гораздо лучше улавливают грамматику, чем смысл.
- GPT хорошо доучивается на задачи условной генерации текста:
  - Ответная реплика в диалоге.
  - Выжимка текста.
- Нельзя ли так же доучить на генерацию правильных предложений по исходным?

GPT и грамматические ошибки

- Нейронные модели достигли больших успехов в генерации текста.
- При этом они гораздо лучше улавливают грамматику, чем смысл.
- GPT хорошо доучивается на задачи условной генерации текста:
  - Ответная реплика в диалоге.
  - Выжимка текста.
- Нельзя ли так же доучить на генерацию правильных предложений по исходным?
  - Пока не получается!
  - GPT слишком свободно порождает текст, изменения вносятся не только в грамматику, но и в смысл.

Генерация искусственных данных

- Обучение чаще всего проводится в основном на синтетических данных:
  - Искусственное внесение шума.
  - Обратный перевод (возможно, зашумлённый).

Генерация искусственных данных

- Обучение чаще всего проводится в основном на синтетических данных:
  - Искусственное внесение шума.
  - Обратный перевод (возможно, зашумлённый).
- От их качества зависит качество получившейся модели.
- Влияющие факторы:
  - Соотношение “простых” и “сложных” данных.
  - Распределение искусственных данных по типам ошибок.

Исправление грамматических ошибок: применение

- Важно подстроить исправление грамматических ошибок под задачу:
  - Изучение иностранного языка.
  - Проверка работ школьников на родном языке.
  - Индивидуальная обучающая траектория (персонификация).

Исправление грамматических ошибок: применение

- Важно подстроить исправление грамматических ошибок под задачу:
  - Изучение иностранного языка.
  - Проверка работ школьников на родном языке.
  - Индивидуальная обучающая траектория (персонификация).
- Что считать ошибкой, зависит от сферы применения.
- Нужны размеченные данные для каждой сферы применения (проблема, например, при проверке работ школьников).