Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

КУРСОВАЯ РАБОТА

Исследовательский проект на тему "Аугментация текста для задачи бинарной классификации"

Выполнил студент группы 171, 3 курса: Рак Арина Сергеевна

Руководитель КР: приглашенный преподаватель, Литвинов Денис Владимирович

Москва 2020

Содержание

Аннотация

В данной работе рассматривается задача аугментации текста, относящаяся к области обработки естественного языка, для проблемы бинарной классификации. Хотя эта задача имеет множество решений для изображений (например, сдвиг, вращение, отражение и т. д.), для текстовых данных она является куда более сложной. В этой работе была выбрана проблема классификации комментариев из соревнования "Quora Insincere Questions Classification". На этих данных была обучена базовая модель с качеством, сравнимым с сильными решениями соревнования. К исходным данным были применены различные методы аугментации текста для достижения лучших результатов. Способы аугментации сравнивались по нескольким параметрам: время применения и F_1-мера.

Annotation

This work is dedicated to an NLP task of text augmentation for the problem of binary text classification. Though the task can be easily solved for image data (by rotatiting, shifting or adding noise for the data), for example, it is not so simple when it comes to text data. In this work Quora Insincere Questions Classification competition was chosen as the main problem. A GRU-based architecture was fixed. After that different approaches of data augmentation were applied to the original dataset and compared in inference time and F_1-score, to achieve higher score with the same architecture.

Ключевые слова

Обработка естественного языка, аугментация данных, классификация текста, бинарная классификация, глубинное обучение

1 Введение

Аугментация данных – увеличение выборки данных для обучения через модификацию существующих данных. Задача аугментации данных важна и применима для работы с любым типом данных, особенно когда речь идет о глубинном обучении. Чем сложнее модель, тем больше данных ей необходимо для избежания переобучения. Аугментация данных – подход, позволяющий увеличить обобщающую способность модели за счет увеличения размера датасета без затрат на разметку данных человеком. Говорят, что алгоритм обучения обладает способностью к обобщению, если вероятность ошибки на тестовой выборке достаточно мала или предсказуема, то есть не сильно отличается от ошибки на обучающей выборке.

Существует три основных составляющих глубинного обучения: модель, данные и оборудование выполняющее вычисления. На данный момент, имеются инфраструктурные решения, позволяющие пользоваться лучшими моделями (реализованными в библиотеках) и облачными технологиями, дающими доступ к оборудованию с высокими вычислительными мощностями. Для работы с данными решения такого вида не представляются возможными, но стандартизированные методы аугментации и препроцессинга позволяют улучшить процесс обучения без дополнительных затрат. То есть аугментация данных актуальна для любой задачи обучения с учителем.

Также аугментация данных может рассматриваться как дополнительный способ зашумления данных, например для обучения, шумоподавляющих автоэнкодеров для текстов. Например, Lewis et al. (2019)

В данной работе речь идет об аугментации текста в задаче бинарной классификации. Например, это может быть классификация писем на спам и не спам по их содержанию, или классификация отзывов о товарах на положительные и отрицательные, или комментариев на позитивные и негативные. В таких задачах зачастую классы бывают несбалансированными, что приводит к затруднению обучения модели. Аугментация только миноритарного класса

может решить эту проблему. Аугментация обоих классов решает проблему небольшого объема данных для обучения.

На данный момент одним из лучших методов аугментации текстовых данных является перевод предложения с языка оригинала на другой язык (чаще всего английский, немецкий или французский, так как для этих языков существует множество предобученных моделей) и обратно на язык оригинала. В рамках этой работы я сравниваю различные способы аугментации текста с данным методом.

Цель данной работы – предложить эффективную и быструю аугментацию текста для задачи бинарной классификации. Эффективность и скорость будет сравниваться с результатами для аугментации машинным переводом.

Для этого на задаче бинарной классификации "Quora Insincere Questions Classification" была обучена базовая модель. К исходным данным была применена аугментация переводом, с помощью предобученной модели трансформера. Были проведены эксперименты по аугментации данных с помощью библиотеки nlpaug, как на уровне слов, так и на уровне символов. Все эксперименты проводились для модели с обучаемыми эмбеддингами и модели с предобученными эмбеддингами. Также провелось сравнение влияние аугментации при токенизации на слова и подслова.

Подходы сравнивались по скорости применения и качеству, в зависимости от способа аугментации и доли аугментированных данных.

Все рассмотренные методы дают значительный выигрыш во времени применения (сотни или тысячи раз) по сравнению с переводом, и не требуют работы на видеокарте. При этом как для модели с обучаемыми эмбеддингами, так и для модели с предобученными эмбеддингами, альтернативные подходы смогли показать качество лучшее, чем аугментация переводом.

В главе 3 подробно описывается рассматриваемая задача, данные и их обработка, базовая модель и описывается процесс обучения. В главе 4 описываются все использованные методы аугментации, приводятся сравнение результатов и времени работы.

2 Обзор литературы

Для аугментации данных нужно выполнить преобразование над элементами исходного набора данных, инвариантные к ответам для решаемой задачи. (например, положительный комментарий должен остаться положительным в задаче классификации комментариев на позитивные и негативные).

В работе He et al. (2019) было показано, что эмпирический риск классификатора, обученного на аугментированыых данных, является верхней оценкой на ожидаемый риск классификатора, обученного на исходных данных. При этом точность этой оценки зависит от размера аугментационной выборки и того, насколько меняется совместное распределение объект-ответ после аугментации.

С вероятностью $1 - \delta$ выполняется:

$$R(f_{aug}|P) \leqslant \hat{R}(f_{aug}|P) + O\left(\left(\frac{|F|_{VC} - log\delta}{N \cdot M}\right)^{\alpha}\right) + \varepsilon_2 - \varepsilon_1, \qquad (1)$$

где P совместное распределение объектов (x) и ответов (y).

$R(f|P)$ ожидаемый риск классификатора f, $\hat{R}(f|P)$ эмпирический риск, $|F|_{VC}$ конечная размерность Вапника-Червоненкиса (константа, характеризующая классификатор), N размер начальной выборки, $M \cdot N$ размер аугментированной выборки, $0.5 \leqslant \alpha \leqslant 1$

$R(f_{aug}|P_{aug}) - R(f_{aug}|P) = \varepsilon_1 \geqslant 0$

$\hat{R}(f_{aug}|P_{aug}) - \hat{R}(f_{aug}|P) = \varepsilon_2 \geqslant 0$

Это дает основания полагать, что обучение на аугментированных данных, действительно помогает лучше решать исходную задачу.

Одним из самых простых спобосов аугментации текстовых данных является замена r случайных слов синонимами из Thesaurus, предложенная Zhang et al. (2016) и Wei et al. (2019) (во второй статье были добавлены стоп-слова, которые не подлежат замене и удалению), а также добавление случайного синонима случайного слова на случайную позицию предложения, удаление

слова из предложения или перестановка двух случайных слов в предложении. Все эти способы характеризуются вероятностью p аугментации конкретного слова, где p является гиперпараметром.

Также для аугментации можно заменять слова на близкие не только по словарю синонимов, но и на ближайших соседей в пространстве эмбеддингов (векторных представлений токенов) Yang Wang et al. (2015) или использовать контекстно зависимые эмбеддинги Devlin et al. (2019).

Языковые модели также применяются для аугментации текста. Например, Fadaee et al (2017) использовали ее для задачи перевода на редкие языки с маленьким объемом данных. Языковая модель обучалась на языке оригинала, далее редкое слово s выкидывалось из контекста и заменялись словом $\hat{s}$, предложенными моделью в оригинальном языке. В редком языке соответствующее слово t заменялось переводом $\hat{s} : \hat{t}$.

Популярным является подход преобразования текста переводом, предложенный Sennrich et al. (2016) для задач машинного перевода. Этот подход был обобщен с помощью перевода с языка оригинала на промежуточный язык и обратно на язык оригинала, что позволяет сохранить семантику, но, возможно, изменить строение предложения или используемые слова. Именно этот метод взят за основу в данной работе.

Альтернативным подходом, является аугментация текста на уровне символов, описанная, например, в Coulombe (2018), когда к исходному тексту добавляется "шум", с помощью замены символов на близкие, удаления или добавления случайных символов, смены регистра символов и изменения пунктуации.

В рамках данной работы планируется сравнить методы замены слов синонимами Wei, Zou (2019), перестановки и удаления случайного слова, аугментации на уровне символов добавлением шума Coulombe (2018) и по визуальной близости, с помощью библиотеки nlpaug, а также изучить способы аугментации с помощью перевода Sennrich et al. (2016).

3 Задача и базовое решение

3.1 Решаемая задача

Для исследования аугментации текста была выбрана задача классификация комментариев на "токсичные"и не "токсичные"из соревнования "Quora Insincere Questions Classification".

В соревновании предоставлен набор данных с вопросами, заданными на Quora на английском языке, и метками о том, являются эти вопросы "токсичными" или нет. Метки содержат некоторое количество шума.

Примерами "токсичных" вопросов являются: 'Why are software engineers on Quora so judgmental?' или 'Do Indian Muslim women envy Hindu women?'.

Тренировочная часть набора данных содержит 1.3 миллиона вопросов разной длины (Рис. 1), ≈6.8% которых являются "токсичными". То есть классы являются сильно несбалансированными.

В соревновании метрикой качества выбран $F_1 - score = \frac{2 \cdot precision \cdot recall}{precision + recall}$, что логично при работе с несбалансированной выборкой, поэтому для сравнения методов аугментации в данной работе также использовалась эта метрика.

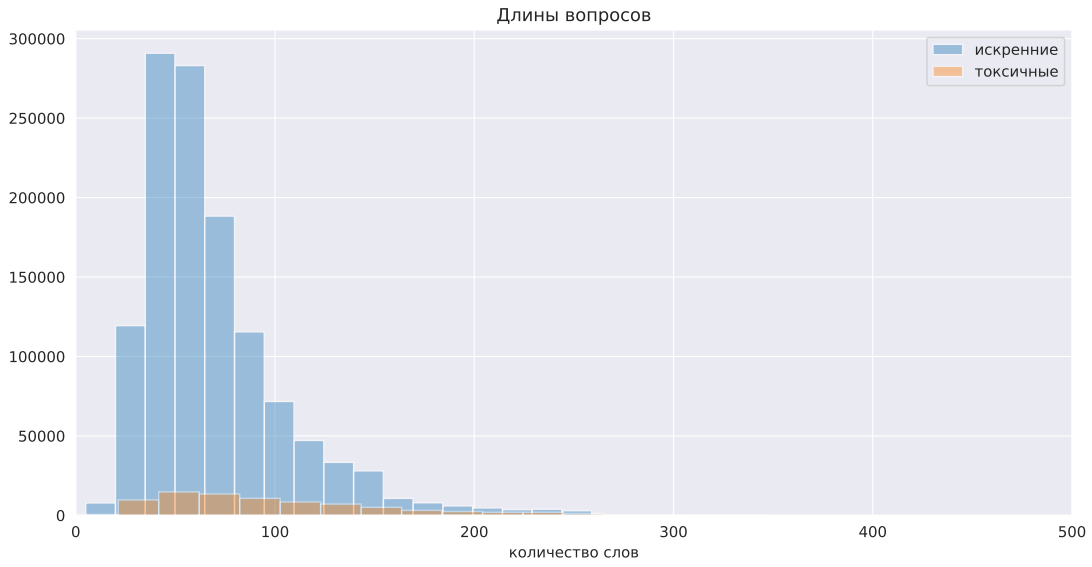


Рис. 3.1: Гистограмма длин вопросов в исходном наборе данных

В соревновании также представлены 4 вида эмбеддингов размерности 300:

- GoogleNews-vectors-negative300
- glove.840B.300d
- wiki-news-300d-1M
- paragram_300_sl999

В ходе работы была реализована модель, работающая как с предобученными эмбеддингами, так и с обучаемыми в процессе эмбеддингами. Все эксперименты проводились для обеих ситуаций.

В качестве предобученных эмбеддингов использовались сконкатенированные пары из предоставленных в соревновании. Бейзлайн был обучен на всех возможных парах. Лучший результат был получен для конкатенации GloVe и GoogleNews word2vec, поэтому в дальнейшем использовались только эти эмбеддинги.

3.2 Базовое решение

3.2.1 Препроцессинг

Во время препроцессинга текст переводится в нижний регистр, пунктуация окружается пробелами (чтобы рассматривать знаки препинания как отдельные токены), все цифры заменяются на символ "#", а пропуски заменяются на символ "_##_". Далее обработанные вопросы токенизируются. Была рассмотрена токенизация как по словам, так и по подсловам. Все последовательности обрезаются до длины в 50 токенов, для ускорения обучения.

Для токенизации по подсловам было использовано кодирование байтовых пар (BPE. Gage, 1994) — алгоритм сжатия, зачастую используемый для токенизации при решении задачи машинного перевода Sennrich et al. (2016). Это итеративный метод токенизации. Изначально словарь состоит из символов алфавита. Далее выписываются частоты пар элементов словаря, встречаемых в тексте, и самая частотная пара добавляется в словарь. Такая процедура повторяется фиксированное количество итераций. Для нашей задачи

50000 итераций показали наилучший результат. Кодирование байтовых пар обучается на неаугментированной тренировочной выборке и применяется после аугментации.

При обучении внутри одного батча, к последовательностям применяется паддинг специальными символами <pad>, до длины максимальной последовательности в батче.

3.2.2 Модель

Для базового решения была выбрана модель, основанная на двунаправленной GRU (Рис. 2).

Размер батча (batch_size): 512

Размерность эмбеддингов (emb_dim): 600 (конкатенация пары эмбеддингов из предложенных в соревновании)

Длина последовательности (seq_len) равна максимальной длине последовательности внутри этого батча

Размерность модели (model_dim): 128

Модель принимает на вход тензор $batch\_size \times emb\_dim \times seq\_len$.

Далее токены переводятся в эмбеддинги, которые подаются на вход однослойному двунаправленному управляемому реккурентному блоку (BiGRU). Если эмбеддинги обучаемые, то перед подачей на вход GRU, они проходят через слой дропаута (p=0.45).

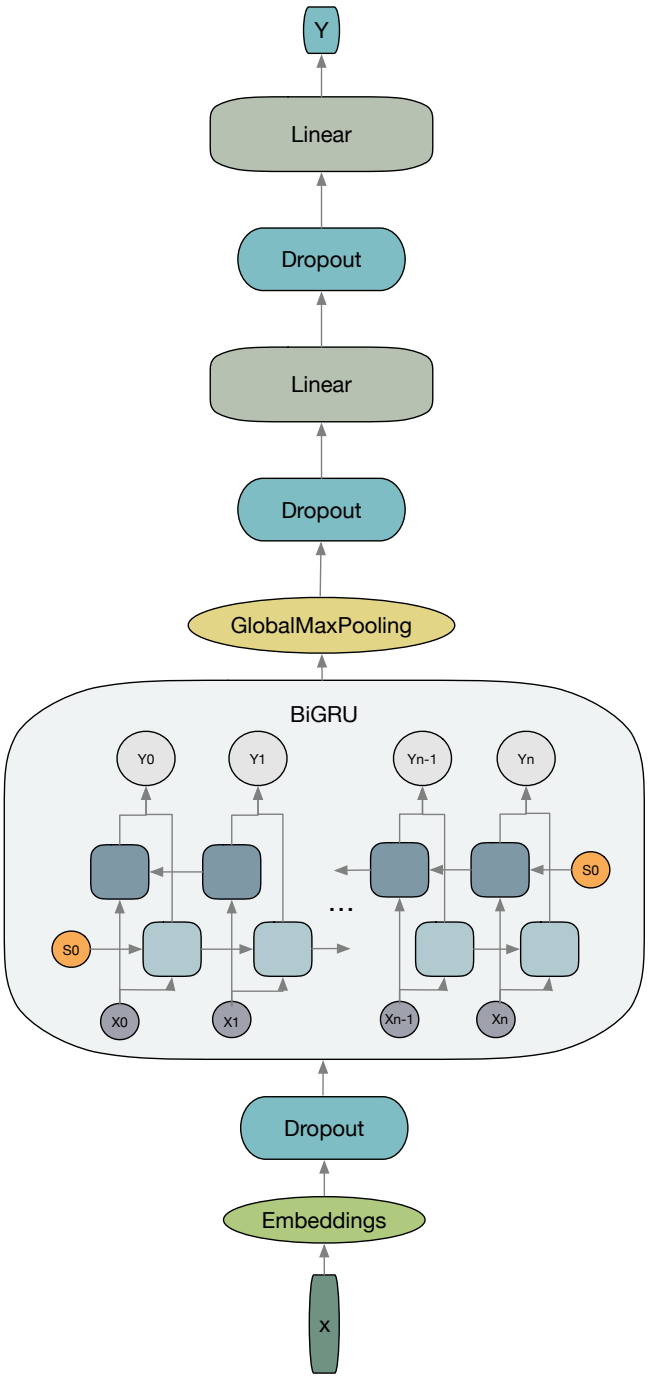


Рис. 3.2: Модель базового решения

Рисунок 3.3 показывает $F_1 - score$ для различных гиперпараметров мо-

дели бейзлайна, где $emb\ ij$ соответсвует используемой паре эмбеддингов, а d соответствует коэффициенту дропаута. На графике можно заметить, что использованные для базовой модели параметры являются оптимальными.

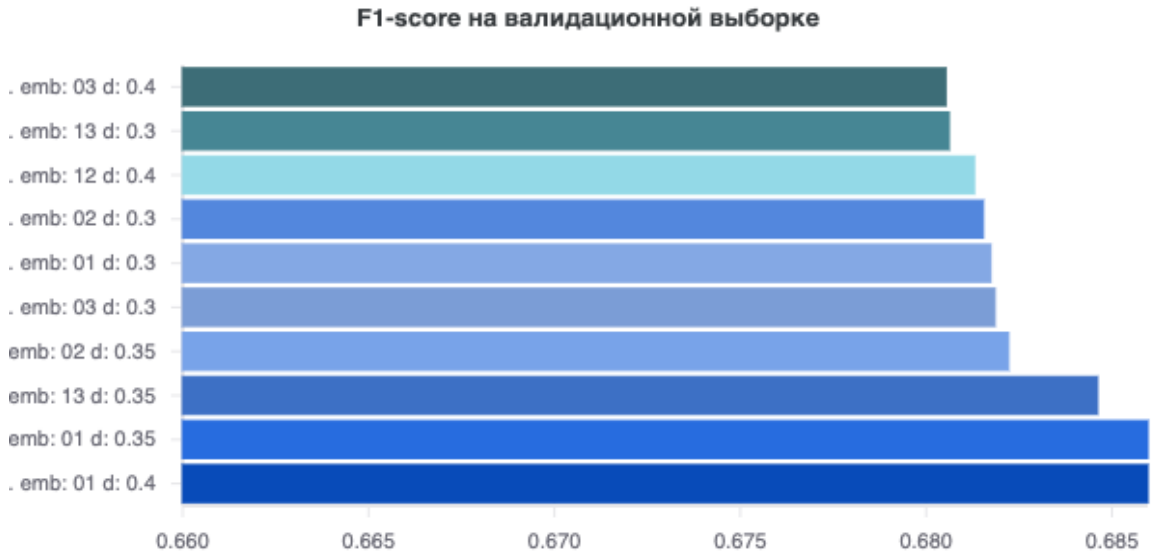


Рис. 3.3: Аугментация добавлением символа

К конкатенированным по направлениям выходам ($batch\_size \times seq\_len \times model\_dim \cdot 2$), применяется глобальный макспуллинг ($batch\_size \times model\_dim \cdot 2$) и дропаут (p = 0.35).

После этого к выходам применяется линейный слой, не меняющий размерности, затем дропаут (p = 0.35) и еще один линейный слой ($batch\_size \times 1$), переводящий его в "скор токсичности". В качестве функции активации везде используется ReLU().

Для получения вероятности класса, к выходу модели применяется сигмоида.

3.2.3 Обучение

Данные были разбиты на обучающие и тестовые в пропорции 7 : 3, а обучающие данные, в свою очередь были разбиты на валидационные и обучающие в пропорции 1 : 9.

Аугментируется только обучающая выборка. При этом ее размер увеличивается в 2 раза.

Для изучения аугментации обучается две базовых модели, одна с предобученными эмбеддингами (сконкатенированные GloVe и FastText, предложенные в соревновании), а вторая с обучаемыми эмбеддингами.

Первая модель достигает $F_1 - score = 0.685$, а вторая $F_1 - score = 0.6483$

4 Аугментация

В этой главе будут описаны иследованные в ходе работы методы аугментации. Для иллюстрации их работы, каждый метод будет применен к примеру из обучающей выборки: "what are the real advantages of using quora ?"

4.1 Аугментация переводом

В качестве первого подхода аугментации была выбрана аугментация переводом предложений с английского на немецкий и обратно с немецкого на английский.

Для перевода была использована предобученная на датасете WMT16 модель трансформер Vaswani et al (2017) из библиотеки fairseq.

Преобразование исходного примера: "what are the real **benefits** of using quora?"

Каждый из последующих методов задается гиперпараметром p: долей слов в предложении, которые будут аугментированы. Поэтому сначала производится сравнение метода с разными гиперпараметрами, а потом сравнение лучших представителей методов между собой. Во всех экспериментах случайность была зафиксирована, что ведет к воспроизводимости результатов. Для каждого метода будут представлены результаты для разных гиперпараметров, в сравнении с базовой моделью и аугментацией переводом.

4.2 Аугментация на уровне слов

4.2.1 Аугментация синонимами

В данном подходе случайное слово из предложения заменяется на случайный синоним с вероятностью p. Синонимы выбираются из лексической базы данных английского языка WordNet, содержащей 117 000 групп синонимов.

Преобразование исходного примера при p=0.35 : "what **be** the real advantage of **use** quora ?"

Результаты для различных гиперпараметров представлены на рисунке 7.1, см. Приложения.

4.2.2 Аугментация случайной перестановкой

В данном подходе аугментация происходит за счет перестановки случайных слов в предложении с вероятностью p

Преобразование исходного примера при p=0.35: "what **the are** real **of advantages quora using** ?"

Результаты для различных гиперпараметров представлены на рисунке 7.2 см. Приложения.

4.2.3 Аугментация случайным удалением

В данном подходе аугментация происходит за счет удаления случайных слов в предложении с вероятностью p

Преобразование исходного примера при p=0.35: "are the advantages of using ?"

Результаты для различных гиперпараметров представлены на рисунке 7.3, см. Приложения.

$F_1 - score$ для описанных методов, в зависимости от выбора эмбеддингов и препроцессинга, представлен в Таблице 4.1. Можно заметить, что каждый из представленных способов аугментации улучшает качество, по сравнению

с базовой моделью. При этом результат аугментации переводом не всегда оказывается наилучшим.

Таблица 4.1: F_1-score на тестовой выборке для аугментации на уровне слов

	Базовая модель	Синонимы	Перестановка слова	Удаление слова	Перевод
Предобученные эмбеддинги	0.685	0.6853	0.**6873**	0.6865	0.6861
Обучаемые эмбеддинги	0.6483	**0.651**	0.649	0.6494	0.6504
BPE + обучаемые эмбеддинги	0.6481	0.6499	0.6509	0.6501	**0.6517**

Важно отметить, что аугментация переводом является времязатратной даже при использовании предобученной модели. Для перевода тренировочной выборки с английского на немецкий и с немецкого на английский с помощью предобученной модели трансформера потребовалось 84,3 часа обучения на видеокарте (GPU), без видеокарты это заняло бы 1000 часов. Также она требует в два раза больше памяти, чем остальные методы, так как требуется хранить исходную выборку, перевод на немецкий и перевод на английский. Для применения остальных методов, рассматриваемых в данной работе, видеокарта не требуется. Длительность применения аугментации на уровне слов, в зависимости от доли аугментированных слов, отражена в Таблице 4.2. Аугментация с помощью синонимов выгрывает в 250 раз во времени, для остальных методов речь идет о выигрыше в 1000 и больше раз.

4.3 Аугментация на уровне символов: учет опечаток

Для того, чтобы учесть возможные опечатки или исправить уже существующие, было использовано два вида аугментации: аугментация OCR и keyboard аугментация.

Таблица 4.2: Время аугментации, секунды

	Синонимы	Перестановка слова	Удаление слова	Перевод
0.1	1137	114	103	-
0.2	1234	130	107	-
0.3	1292	146	112	-
0.4	1360	161	116	-
0.5	1249	174	120	-
0.6	1125	183	122	-
Среднее	1216	151	113	**303649** на GPU **3600000** на CPU

4.3.1 Аугментация OCR

OCR (optical character recognition) — оптическое распознавание символов. Аугментация ocr подразумевает замену случайных символов на визуально близкие. Например, i на 1 или o на 0. Доля аугментированных слов определяется гиперпараметром p. Чаще всего, этот вид аугментации применяется при классификации текста, распознаного по изображению (например, рукописного) тогда схожие визуально символы, не подходящие по смыслу, могут появиться в исходных данных.

Преобразование исходного примера при p=0.35: "what are the real **advanta9e8** of **usin9** quora ?"

Результаты для различных гиперпараметров представлены на рисунке 7.4, см. Приложения.

4.3.2 Аугментация keyboard

Аугментация для симуляции опечаток и удаления существующих, заменой случайных символов на близкие по расположению на клавиатуре. Доля аугментированных слов определяется гиперпараметром p.

Преобразование исходного примера при p=0.35: "what are the **reAl** advantages of using quora ?"

Результаты для различных гиперпараметров представлены на рисунке

7.5, см. Приложения.

В Таблице 4.3 видно, что эти методы также являются эффективными методами аугментации, но менее сильными, чем аугментация переводом. Каждый из подходов применяется за небольшое количество времени, что отражено в Таблице 4.4. Также можно заметить, что чем больше доля аугментированных слов, тем больше занимает аугментация.

Таблица 4.3: F1-score на тестовой выборке, для аугментации на уровне символов

	Базовая модель	Оптически близкие символы	Близкие на клавиатуре символы	Перевод
Предобученные эмбеддинги	0.685	0.686	0.6828	**0.6861**
Обучаемые эмбеддинги	0.6483	0.6491	0.6502	**0.6504**
BPE + обучаемые эмбеддинги	0.6481	0.649	0.6491	**0.6517**

Таблица 4.4: Время аугментации, секунды

	Оптически близкие символы	Близкие на клавиатуре символы	Перевод
0.1	173	168	-
0.2	201	194	-
0.3	232	221	-
0.4	261	246	-
0.5	287	269	-
0.6	304	283	-
Среднее	243	230	**303649** на GPU **3600000** на CPU

4.4 Аугментация на уровне символов: добавление шума

Также для аугментации можно вносить шум, в существующие предложения. Стоит отметить, что внесение шума почти всегда приводит к несуществующим словам, что в свою очередь приведет к большему количеству слов, не содержащихся в словаре и не имеющих предобученного векторного представления. Таким образом для этих методов аугментации больше подходит модель с обучаемыми эмбеддингами.

4.4.1 Аугментация добавлением случайного символа

Аугментация добавлением случайного символа в случайное слово, доля аугментированных слов задается гиперпараметром p.

Преобразование исходного примера при p=0.35: "what are the real advantages of **usihng qzuora** ?"

Результаты для различных гиперпараметров представлены на рисунке 7.6, см. Приложения.

4.4.2 Аугментация перестановкой случайного символа

Аугментация перестановкой случайных символов случайного слова, доля аугментированных слов задается гиперпараметром p.

Преобразование исходного примера при p=0.35: "what are the real **avdnatgaes** of using quora ?"

Результаты для различных гиперпараметров представлены на рисунке 7.7, см. Приложения.

4.4.3 Аугментация удалением случайного символа

Аугментация удалением случайного символа из случайного слова, доля аугментированных слов задается гиперпараметром p.

Преобразование исходного примера при p=0.35: "what are the **rel advaaes** of using quora ?"

Таблица 4.5: F_1-score на тестовой выборке, для аугментации на уровне символов

	Базовая модель	Добавление символа	Перестановка символа	Удаление символа	Перевод
Предобученные эмбеддинги	0.685	0.6844	0.6838	0.6838	**0.6861**
Обучаемые эмбеддинги	0.6483	0.6482	0.6488	0.6498	**0.6504**
BPE + обучаемые эмбеддинги	0.6481	0.6512	0.6492	0.6489	**0.6517**

Результаты для различных гиперпараметров представлены на рисунке 7.8, см. Приложения.

В Таблице 4.5 представлены результаты аугментации добавлением шума. Заметим, что для предобученных эмбедингов внесение шума таким образом не является корректным способом аугментации. Шум приводит к увеличению слов вне словаря, что выражается в ухудшении качества по сравнению с базовой моделью. Для внедрения шума с предобученными эмбедингами можно, например, использовать дропаут после слоя эмбединга.

В Таблице 4.6 можно заметить, что эти способы также являются быстрыми.

Таблица 4.6: Время аугментации, секунды

	Добавление символа	Перестановка символа	Удаление символа	Перевод
0.1	165	170	158	-
0.2	186	198	172	-
0.3	208	226	187	-
0.4	229	253	201	-
0.5	229	279	214	-
0.6	246	295	222	-
Среднее	216	245	192	**303649** на GPU **3600000** на CPU

5 Заключение

В ходе данной работы, был проведен ряд экспериментов, сравнивающих различные методы текстовой аугментации для задачи бинарной классификации. Целью работы было найти альтернативу аугментации переводом, являющуюся эффективным методом, требовательным как к памяти, так и времени. По результатам экспериментов методы аугментации на уровне слов были признаны наиболее удачными. В частности, перестановка случайного слова и аугментация с помощью синонимов, во всех случаях давали прирост качества. Также необходимо отметить, что эти методы чувствительны к гиперпараметрам и ухудшают качество по сравнению с бейзлайном при неправильном выборе доли аугментированных слов, что отражено на Рисунке 7.1 и Рисунке 7.2. Наименее эффективными оказались методы перестановки и удаления символа, которые не проявляют значительную чувствительность к гиперпараметрам, но применение которых, в большинстве случаев приводит к ухудшению качества классификации. Также было эксперементально подтверждено, что аугментация на уровне символов не подходит для использования с предобученными эмбеддингами.

Интересным направлением для дальнейшего исследования является аугментация по словам близким в пространстве эмбеддингов и контекстно зависимых эмбеддингов. Эти способы не были опробованы в данной работе из-за технических ограничений, связаных с тем, что они требовательны к памяти. Альтернативным способом аугментации для изучения является обогащение текстового классификатора с помощью графа знаний Annervaz et al. (2018), на которое будет направлено предстоящее исследование.

6 Список литературы

1. Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer.
   BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint arXiv:1910.13461, 2019.*

2. Zhuoxun He, Lingxi Xie, Xin Chen, Ya Zhang, Yanfeng Wang, Qi Tian.
   Data Augmentation Revisited: Rethinking the Distribution Gap between Clean and Augmented Data. *arXiv preprint arXiv:1909.09148, 2019.*

3. Xiang Zhang, Junbo Zhao, Yann LeCun.
   Character-level Convolutional Networks for Text Classification. *In Proceedings of Advances in Neural Information Processing Systems 28 (NIPS 2015).*

4. Jason Wei, Kai Zou.
   EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *arXiv preprint arXiv:1901.11196, 2019.*

5. William Yang Wang, Diyi Yang.
   That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015.*

6. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova.
   BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805, 2019.*

7. Marzieh Fadaee, Arianna Bisazza, Christof Monz.
   Data Augmentation for Low-Resource Neural Machine Translation. *arXiv preprint arXiv:1705.00440, 2017.*

8. Rico Sennrich, Barry Haddow, Alexandra Birch.
Improving Neural Machine Translation Models with Monolingual Data. *arXiv preprint arXiv:1511.06709, 2015.*

9. Claude Coulombe.
IText Data Augmentation Made Simple By Leveraging NLP Cloud APIs. *arXiv preprint arXiv:1812.04718, 2018.*

10. Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hananneh Hajishirzi.
BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION. *arXiv preprint arXiv:1611.01603, 2018.*

11. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin.
Attention Is All You Need. *In Proceedings of Advances in Neural Information Processing Systems (NIPS 2017).*

12. Rico Sennrich and Barry Haddow and Alexandra Birch.
Neural Machine Translation of Rare Words with Subword Units. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016).*

13. Ivan Provilkov, Dmitrii Emelianenko, Elena Voita.
BPE-Dropout: Simple and Effective Subword Regularization. (2019)

14. Annervaz K M, Somnath Basu Roy Chowdhury, Ambedkar Dukkipati. Learning beyond datasets: Knowledge Graph Augmented Neural Networks for Natural language Processing. (2018)

7 Приложения

Исходный код: github.com/textaugmentation

Визуализации всех экспериментов доступны по следующим ссылкам:

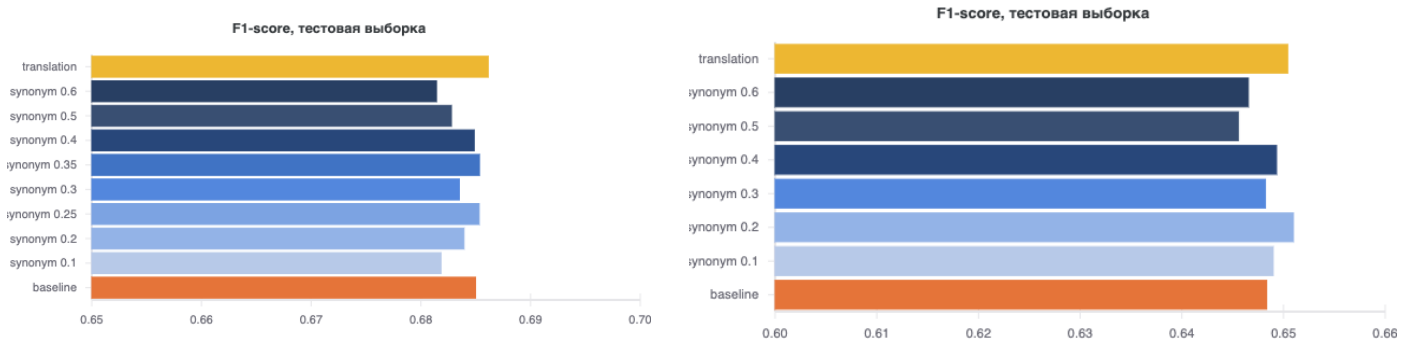
Предобученные эмбеддинги

Обучаемые эмбеддинги

Обучаемые эмбеддинги + BPE

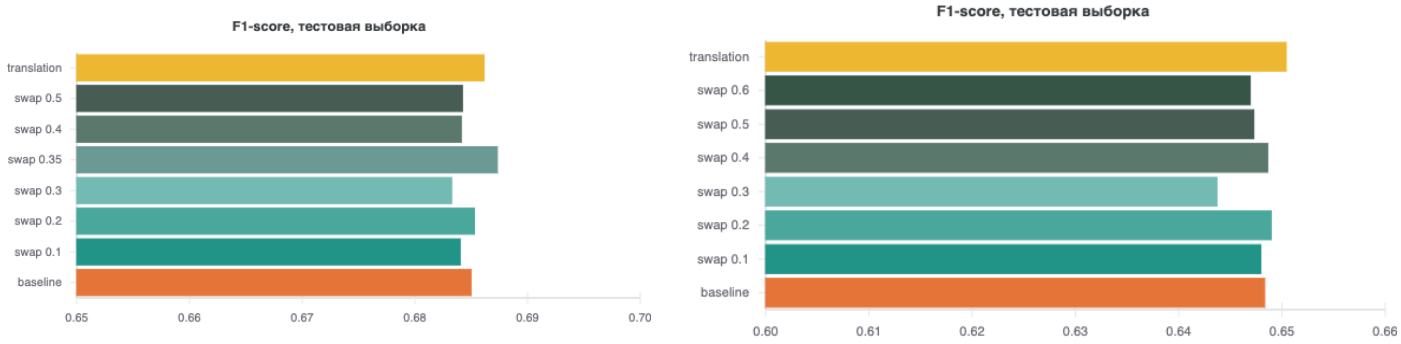
Суммарное время вычислений ≈ 450 часов

Число проведенных экспериментов: ≈ 200



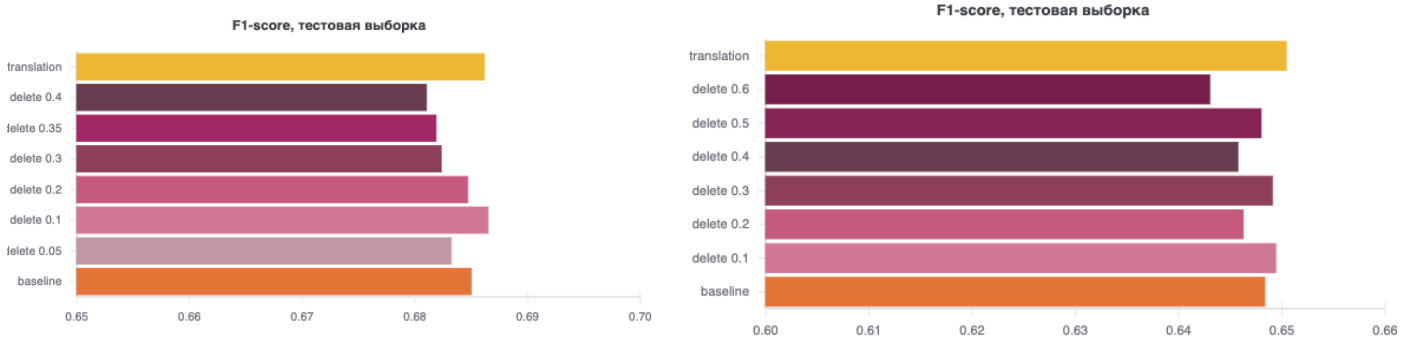
(a) Предобученные эмбеддинги (b) Обучаемые эмбеддинги

Рис. 7.1: Аугментация синонимами



(a) Предобученные эмбеддинги (b) Обучаемые эмбеддинги

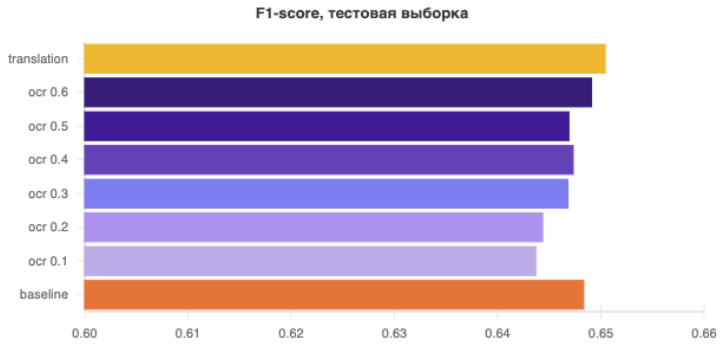
Рис. 7.2: Аугментация перестановкой



(a) Предобученные эмбеддинги (b) Обучаемые эмбеддинги

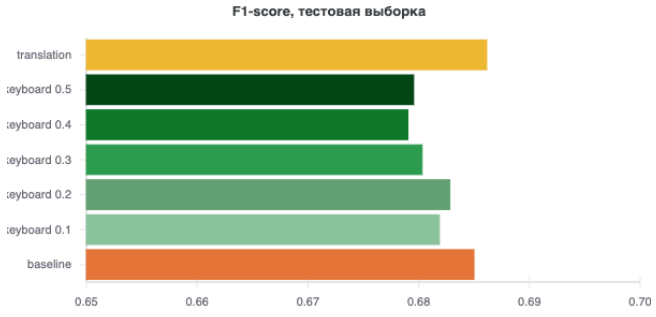
Рис. 7.3: Аугментация удалением

(a) Предобученные эмбеддинги

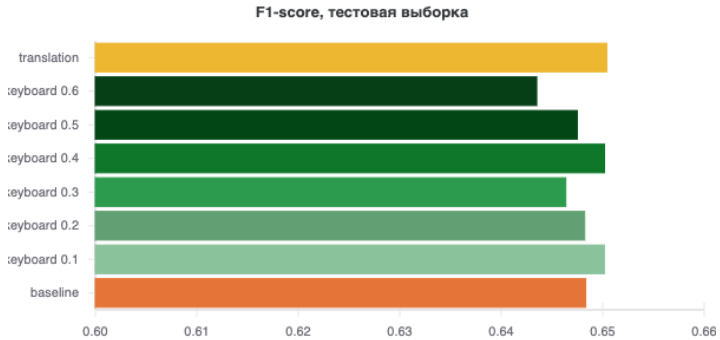


(b) Обучаемые эмбеддинги

Рис. 7.4: Аугментация ocr



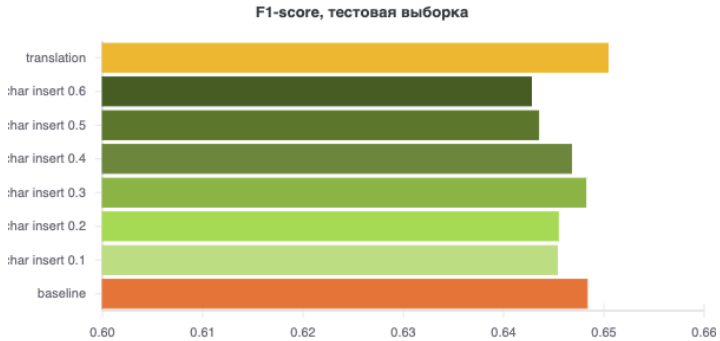
(a) Предобученные эмбеддинги



(b) Обучаемые эмбеддинги

Рис. 7.5: Аугментация keyboard

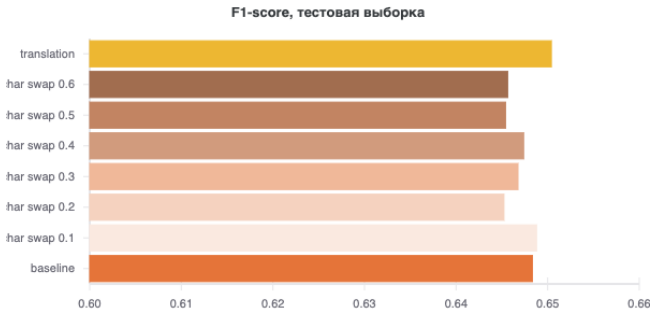
(a) Предобученные эмбеддинги



(b) Обучаемые эмбеддинги

Рис. 7.6: Аугментация добавлением символа

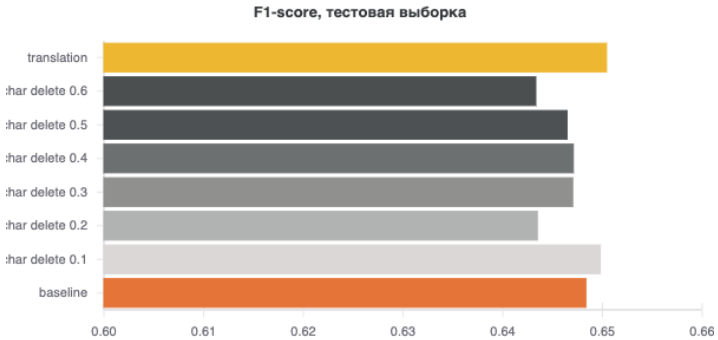
(a) Предобученные эмбеддинги



(b) Обучаемые эмбеддинги

Рис. 7.7: Аугментация перестановкой символа

(a) Предобученные эмбеддинги



(b) Обучаемые эмбеддинги

Рис. 7.8: Аугментация удалением символа