Министерство науки и высшего образования Российской Федерации
ФГАОУ ВО «Волгоградский государственный университет»
Институт Математики и информационных технологий
Кафедра компьютерных наук и экспериментальной математики

Научно-исследовательская работа

Генерация грамматических заданий для изучения Английского языка при помощи NLP

Выполнил:
Косенко
Дмитрий Павлович
Студент 2 курса
группы МОС-191

Научный руководитель:
Клячин
Владимир Александрович
д.ф.-м.н., зав. кафедрой
КНЭМ

Волгоград 2021

Содержание

1 Введение

Область онлайн изучения иностранных языков претерпела колоссальный рост в период 2019-2021 года. Первой причиной является усиление глобализации, рост количества пользователей смартфонов и предпочтение крупных корпораций сотрудников, владеющих несколькими языками. Все эти факторы в совокупности предполагают дальнейший рост рынка с 5 923,90 миллиона долларов США в 2019 году до 12 452,63 миллиона долларов США к 2025 году при CAGR 13,18% [1].

Второй причиной является пандемия COVID-19. Согласно данным UNESCO [2] из-за ограничительных мер, более 1.2 миллиарда студентов пришлось получать образование из дома, что, в свою очередь, многократно повысило нагрузку на учителей.

Проблему перенасыщенности спроса должны помочь решить современные методы обработки естественного языка. В этой работе обозреваются приложения классических для этой области задач таких как: NER, text summarization, text classification и controllable text generation для сферы изучения языков.

2 Сопутствующие работы

Данная работа обозревает возможности приложения различных подходов NLP с разных сторон и решает следующие задачи:

1) grammar constructions detection;
2) controllable text generation;
3) extractive question answering, question generation;
4) text complexity prediction.

2.1 Grammar Constructions Detection

В процессе обучения иностранного языка студент сталкивается с большим количеством теории, связанной с грамматическими конструкциями. Для создания тренировочных упражнений требуются доменные знания от

учителя языка, а также покупка дорогостоящей литературы. У этого подхода есть 2 существенных недостатка. Во-первых, учебник и учитель могут предоставить только ограниченное количество примеров и упражнений, к тому же многие учителя отмечают, что составление большого количества заданий является для них очень утомительным и низко интеллектуальным видом труда. Во-вторых, большая часть примеров для обучения является большей мере искусственной, что, в свою очередь, мешает реальному погружению в современный язык. Также большой проблемой при изучении можно отметить низкий уровень мотивации, эта проблема имеет ряд причин, одной из них является слабое пересечение круга интересов студента и учебного материала.

Изучая данный вопрос, я обнаружил низкую освещенность данной проблемы. Работы из этой области полагаются на алгоритмы с вручную созданными признаками [3, 4].

Данные алгоритмы работают только на эталонных примерах предложений и требуют большого количества обработки различных исключений, частных случаев в языке и глубоких доменных знаний от создателя данных правил.

2.2 Controllable Text Generation

Одной из главных проблем современного образования является списывание. Причиной тому служат открытые сервисы с решебниками и специализированные порталы, где можно купить или попросить решение задания [5, 6]. Одним из решений этой проблемы является создание системы генерации индивидуальных заданий отдельно для каждого ученика. В качестве такой системы может быть использована большая языковая модель.

Формально задача генерации текста языковыми моделями обозначается следующим способом. Мы имеем предтренированную языковую модель p_θ. Эта языковая модель выучила некоторое распределение токенов, путем оптимизации задачи для предсказания следующего токена:

$$L_{LM} = \quad -\sum_{t} log p_{\theta}(x_t | x_{<t})$$

Решение задачи controllable text generation состоит в нахождении стратегии для управления вероятностями для следующего токена модели. Существует 2 основных подхода к этой задаче:

1) изменение метода получения вероятностей следующего токена без какой либо модификации исходных весов модели [31, 41],
2) fine-tuning исходных весов модели под конкретный корпус текста [31].

2.3 Extractive question answering, Question Generation

Задания на понимание прочитанного (reading comprehension), также являются важным типом заданий при изучении языка, так как они улучшают общее ориентирование в материале и формируют высокоуровневое понимание. Данный формат используется в экзаменах формата TOEFL [7], IELTS [8], Duolingo English Test [9], что подтверждает его эффективность для проверки соответствующих навыков. Для подготовки к этим экзаменам требуется большое количество методических пособий с ограниченным количеством примеров. Данную проблему должны решить следующие подходы: extractive question answering, question generation.

Extractive question answering очень обширная тема, здесь я рассмотрю примеры, которые используют датасет SQuAD 1.1 [10].

Существует 2 основных подхода для решении задачи extractive question answering: 1) предсказание начала и конца ответа, который находится в тексте на основе векторного представления вопроса и отрывка 2) sequence-to-sequence подход предсказания целого ответа, подобно задаче перевода.

На текущий момент наиболее качественные и практикоориентированные SoTa модели, которые используют первый подход, являются трансформеры вида BERT [11], RoBERTa [12] и XLNet [13]. Для второго подхода чаще всего используют T5 [14, 15] и GPT-2 [16].

Задача question generation, подразумевает под собой генерацию вопроса на основе отрывка текста и ответа на данный вопрос. Более старые подходы предполагали генерацию на основе синтаксических парсеров и четко написанных правил формирования необходимых видов вопросов [17]. Наиболее современные подходы представляют собой генерацию вопросов как sequence-to-sequence задачу, с использованием моделей ProphetNet [18, 19] и T5 [20, 21].

2.4 Text Complexity Prediction

Во время изучения языка для соблюдения баланса сложности и новизны материала, учителю постоянно требуется, индивидуально для каждого ученика, подбирать новые тексты соответствующего языкового уровня по стандарту CEFR [22].

Частично эта проблема решается специальными методическими пособиями, где тексты специально подобраны по уровням, но такой подход имеет ряд существенных недостатков, о которых говорилось в секции [2.1].

Решение данной проблемы имеет 2 основных подхода: 1) использование лингвистических свойств текста и последующий их анализ при помощи SVM [23, 24] 2) использование нейросетевых моделей, трансформеров вида BERT [24].

3 Методы

3.1 Grammar Constructions Detection

Мой изначальный метод использовал библиотеку spacy для поиска языковых паттернов на основе POS-tags и различных специфических признаков. Данный подход потребовал глубоких доменных знаний от эксперта для обработки исключений и формирования взаимооднозначных паттернов для каждой грамматической конструкции. Данный метод работал аналогично, рассмотренным мною ранее и имел идентичные недостатки. Мой

текущий метод предлагает адаптировать классическую задачу Named Entity Recognition для идентификации грамматических конструкций.

3.11 Grammar constructions dataset

На данный момент не существует общедоступных датасетов с размеченными грамматическими конструкциями на английском языке. В связи с этим было размечено 701 предложение в формате схожем CoNLL-2003 [25]. Данный датасет носит исследовательский характер и был создан исключительно для изучения возможностей данного подхода.

3.12 Data format

Каждый объект в данном датасете является независимым предложением, которое было предварительно разбито токенизатором предоставляемым пакетом spacy [26]. Затем каждый токен был помечен одним из 25 специальных тегов, для разметки использовался инструмент Labelbox [27]. Ниже приведен полный список тегов и несколько примеров из датасета. Метка *I-* означает, что токен содержится внутри конструкции, метка *B-* означает, что токен является началом конструкции.

- O – тег, обозначающий, что токен не относится ни к какому классу;

- a1_be_have_do_in_the_past - тег, обозначающий глаголы "be", 'have', 'do' в их прошедшей форме (past);

- a1_can - тег, обозначающий наличие модального глагола 'can';

- a1_comparative_exept - тег, обозначающий прилагательное в сравнительной форме (исключение);

- a1_comparative_long - тег, обозначающий составное прилагательное в сравнительной форме;

- a1_comparative_short - тег, обозначающий короткое прилагательное в сравнительной форме;

- a1_future_simple - тег, обозначающий наличие грамматической формы будущего простого времени (future simple);

- a1_have_has_got - тег, обозначающий наличие конструкции 'have got' или

'has got';

- a1_past_simple_irreg - тег, обозначающий грамматическую конструкцию прошедшего простого времени (past simple) с неправильным глаголом;

- a1_past_simple_reg - тег, обозначающий грамматическую конструкцию прошедшего простого времени (past simple) с правильным глаголом;

- a1_possesive_s_sing - тег, обозначающий притяжательный апостроф "'s" для существительных в единственном числе;

- a1_possessive_s_plurar - тег, обозначающий притяжательный апостроф "'s" для существительных во множественном числе;

- a1_present_continuous_act_rn - тег, обозначающий грамматическую конструкцию настоящего длительного времени (present continuous) в значении совершения действия в момент речи;

- a1_present_simple_3d_pers - тег, обозначающий грамматическую конструкцию настоящего простого времени (present simple), где подлежащее выражается формой третьего лица единственного числа;

- a1_present_simple_reg_act - тег, обозначающий грамматическую конструкцию настоящего простого времени (present simple) в значении совершения регулярных действий;

- a1_special_questions - тег, обозначающий специальные вопросы;

- a1_superlative_exept - тег, обозначающий прилагательное в превосходной степени (исключение);

- a1_superlative_long - тег, обозначающий составное прилагательное в превосходной форме;

- a1_superlative_short - тег, обозначающий краткое прилагательное в превосходной форме;

- a1_there_is_am_are - тег, обозначающий грамматическую конструкцию "there is" или "there are";

- a1_there_was_were - тег, обозначающий грамматическую конструкцию "there was" или "there were";

- a1_there_will_be - тег, обозначающий грамматическую конструкцию "there will";

- a1_to_be_future_will_be - тег, обозначающий форму глагола "to be" в

будущем времени - "will be";

- a1_to_be_past_was_were - тег, обозначающий форму глагола "to be" в прошедшем времени - "was", "were";

- a1_to_be_present_is_am_are - тег, обозначающий форму глагола 'to be" в настоящем времени - "is", "am", "are";

- a1_want_would_like_to - тег, обозначающий глаголы "want", "would like to" в предложении для выражения желания.

Example №1:

Sentence: Permafrost , which currently sits underneath 80 % of Alaska , is beginning to melt , causing sinkholes and landslides .

Tags: O O B-a1_present_simple_3d_pers I-a1_present_simple_3d_pers I-a1_present_simple_3d_pers O O O O O O O O O O O O O O O O O O

Example №2

Sentence: The wind comes down from the mountain , he added , pointing to the most famous landmark on the Rio skyline .

Tags: B-a1_present_simple_3d_pers I-a1_present_simple_3d_pers I-a1_present_simple_3d_pers I-a1_present_simple_3d_pers O O O O B-a1_past_simple_reg I-a1_past_simple_reg O O O B-a1_superlative_long I-a1_superlative_long I-a1_superlative_long O O O O O O

3.13 Grammar constructions detection model

Я предлагаю модель для детекции грамматических конструкций, основанную на модели RoBERTa base [28] с добавлением линейного слоя размерности H x T в конце модели, где H - hidden size последнего слоя (512), а T количество тегов в датасете (25). Использованы следующие гиперпараметры: learning_rate=5e-5, warmup_steps=300, epochs=10, dropout=0.1, train_batch_size=4.

В результате удалось добиться F1 weighted score для entity level evaluation(ELE) 0.71 и для token level evaluation(TLE) 0.77. Подсчет для ELE производился при помощи пакета seqeval [29], для TLE при помощи sklearn [30].

3.2 Controllable Text Generation

Передо мной стояла задача генерировать обучающие предложения, содержащие специальные грамматические конструкции. На текущий момент существует единственный способ контроля языковой модели, для получения действительно качественных результатов генерации. Это finetuning модели с использованием ключевых слов подобно модели CTRL [31].

3.21 Controllable Text Generation Dataset

Как упоминалось мною ранее в разделе [3.11], не существует датасетов с грамматическими конструкциями на английском языке. А создание больших размеченных датасетов для языкового моделирования является почти невыполнимой задачей. Поэтому было принято использовать модель из раздела [3.13] для автоматической разметки корпуса предложений.

Все тексты были взяты с сайта Medium [32] из 20 тематических разделов на сайте, полный список разделов приложен в разделе Приложение 1. Для эксперимента было обработано 1206 статей. Каждая статья была разбита на независимые предложения, также было сделано грубое допущение, если статья относится к одной из 20 тем, то и предложение из этой статьи относится к этой же теме. Затем была применена модель [3.13] для определения грамматических конструкций в предложениях. Ниже приведено несколько примеров из получившегося датасета.

Example №1:

Sentence: Our brains may only be 3 pounds, however, it's the powerhouse of intelligence, interpretation, initiating body movement, and is the steering wheel that controls all of our behaviours.

Topic: neuroscience

Tenses: a1_to_be_present_is_am_are a1_present_simple_3d_pers

Example №2:

Sentence: Additionally, many Starbucks customers decide to purchase a

coffee because they walk past a store.

Topic: business

Tenses: a1_present_simple_reg_act a1_present_simple_reg_act

3.22 Controllable text generation model

Для контролируемой генерации текста была выбрана оригинальная модель GPT-2 base [33] из пакета Huggingface Transformers [34] без каких либо архитектурных изменений. Для контролируемой генерации все данные были представленные в таком виде: bos_token topic_word sep_token grammar_token_1 grammar_token_2 etc. sep_token original_sentence. В исходный токенизатор были добавлены новые токены:

- bos_token - токен начала строки
- topic_word - одно из 22 ключевых слов представленных в разделе [3.22]
- sep_token - токен разделения сущностей
- grammar_token_1 - токен грамматической конструкции, указывающий на то, что в этом предложении содержится данная конструкция
- original_sentence - оригинальное предложение

Гиперпараметры, которые были использованы для тренировки: batch_size=16, weight_decay=0.01, adam_epsilon=1e-8, epochs=4, warmup_steps=1e2, learning_rate=5e-4.

В результате тренировки было обнаружено, что большая часть генерируемых предложений обладала необходимыми грамматическими свойствами, но отсутствовала яркая выраженность принадлежности к теме.

Ниже приведено несколько примеров на эту тему:

Example №1:

Input sentence: <|BOS|>artificial-intelligence<|SEP|>a1_future_simple<|SEP|>It

Output sentences:

1: It will also give you a sense of the overall impact on your business.

2: It will also help us better understand how to make the decisions we need in order for our company.

3: It will be useful for companies that want to grow their AI systems in ways which enable them more effectively and sustainably.

4: It will use its deep learning model to predict the future and provide predictions based on that.

5: It will make the process of developing and publishing a novel product more efficient.

Example №2:

Input sentence: <|BOS|>artificial-intelligence<|SEP|>a1_can<|SEP|>It

Output sentences:

1: It can also help to identify patterns of behaviour, like when someone is being paid for a certain activity or time.

2: It can take years to build an accurate model of a human being.

3: It can be a lot of work to get started, but you should start by understanding the basics.

4: It can help you decide whether or not to pursue a career in AI.

5: It can be done with a few lines of code, but the output will vary depending on how you want to run it. "I would recommend that we use Python 3 for this task" is probably enough!

Example №3

Input sentence: <|BOS|>startups<|SEP|>a1_to_be_present_is_am_are<|SEP|>We

Output sentences:

1: We are not in the same business as a big tech company, and we may be different from them.

2: We're all in the same boat and we've been through a lot.

3: We are a small team, so it is not always easy to get into the flow of things.

4: We are a team of people who have been in the same business for several years.

5: We are now ready to take the next steps.

6: We're here to help you.

7: We are not a 'technology company'.

3.23 Controllable text generation, meaningful sentences dataset creation

Дальнейшее изучение генерируемых предложений выявило, что все предложения, которые сложно отнести к какой-то теме казались опрашиваемым людям *бессмысленными*. В попытках формализовать данное понятие, касательно нашей задачи, мы пришли к следующим критериям оценки.

Предложение должно:

- быть самостоятельным;
- быть емким и содержать некоторую историю или факт;
- не начинаться с маленькой буквы и не содержать аномальных последовательностей символов, или, иначе говоря, мусора;
- не слишком сильно ссылаться на предыдущие или следующие предложения, чтобы сохранить субъект, о котором говорится в предложении.

Ниже приведены примеры из этого датасета:

Бессмысленные предложения: Consumer advocate Ralph Nader thinks so. | Thus, this causes significant consumer injury. | Therefore, I have got a confession to make. | I think I'm gonna like it here. | When it feels natural, try and focus on these words. | Many will compare it with their salary.

Осмысленные предложения: But don't take my anecdotes as gospel. | The food went from tasting like cardboard to tasting like, well, real pizza. | I don't care where I am, good music calls for a dance party at any moment. | It upset me even more that there are some people who live in their cars in parking lots during the cold winters involuntarily. | He stood in the bedroom blankly for a moment, remembering that day.

В результате создания данного датасета было размечено 3018 предложений из датасета раздела [3.21], 2029 относились к группе осмысленных, 989 к *бессмысленным*. 2018 было отобрано путем стратегии random sampling, а 1000 примеров путем стратегии active learning uncertainty

sampling. При использовании данной стратегии для данной задачи и данного набора данных, существенных различий не обнаружилось.

3.24 Controllable text generation, meaningful sentences model

Для классификации предложений данного датасета была применена модель трансформер RoBERTa base [28]. В качестве признаков для классификации использовались векторные представления токенов [CLS] и [SEP] взятые с предпоследнего слоя модели. В результате тренировки был достигнут следующий F1 Score=0.77

В результате применения данной модели к исходному датасету с сайта medium было отфильтровано примерно 30% всех предложений. Впоследствии исходная модель [3.22] была обучена на новом наборе данных, что существенно улучшило выдаваемые результаты. Пример генерации на новых данных:

Example №1:

Input sentence: <|BOS|>artificial-intelligence<|SEP|>a1_future_simple<|SEP|>I

Output sentence:

1: I will also discuss the implications of this in my next post.

2: I will also mention that in the future, this technique might not be as powerful for all users.

3: I will also use it to make my own annotations of the dataset.

4: I will continue to write about AI and machine learning for the sake of brevity, but first I'll talk a bit more on how it works.

5: I'll leave the AI in a nutshell, but let's take an example.

Example №2:

Input sentence: <|BOS|>artificial-intelligence<|SEP|>a1_can<|SEP|>I

Output sentence:

1: I can't imagine a more exciting time to be working with AI than it is now.

2: I can't see the problem, but I've seen it happen in other industries.

3: I can see how this could lead to a significant decrease in accuracy and precision.

4: I can't imagine how much more complex it would be to have the same amount of data.

5: I can see that, if the algorithm is successful in predicting a user behaviour correctly.

Example №3:

Input sentence: <|BOS|>startups<|SEP|> a1_to_be_present_is_am_are<|SEP|>He

Output sentence:

1: He is passionate about his business and wants to make a difference for the city.

2: He is a founder of the fund, where he invests in venture capital companies.

3: He is passionate about the idea of entrepreneurship and investing in startups.

4: He is passionate about the development of AI and robotics, which are emerging fields.

5: He is a former member of the board at Sequoia Capital and co-founder with Scott Goldstein.

6: He is a big believer in the power of entrepreneurship.

3.3 Extractive question answering, question generation

Для задач extractive question answering, question generation использовался единый датасет SQuAD 1.1 [10] и единая базовая модель.

3.31 Extractive question answering, question generation dataset creation

Так как задачи question answering, question generation являются взаимосвязанными, то было принято решение использовать единый датасет SQuAD 1.1 [10] для обеих задач. В результате EDA анализа [37] было выявлено, что в датасете встречаются выбросы по длине в passage и anwer блоках. Обнаружено что 2707 фрагментов в секции passage имеют длину более чем 350 токенов и 1829 в секции answer длину более чем 25 к общему количеству 87383. В конечном итоге был создан общий датасет для двух задач

описанных выше, который имеет следующий вид:

Extractive question answering

Input sequence: extract answers: _passage_текст_ <hl> _предложение содержащее ответ на вопрос_ <hl> _passage__текст_ </s>

Target sequence: _answer_text_ <sep></s>

Question generation

Input sequence: generate question: _passage__текст_ <hl> _ответ_на_вопрос_ <hl> _passage_текст_ </s>

Target sequence: _question_text_ </s>

3.32 Extractive question answering, question generation model

Для решения этой задачи была использована модель T5-small [35] из пакета Huggingface Transformers [36] без каких либо архитектурных модификаций. В базовый токенизатор были добавлены следующие специальные ключевые токены:

- highlight token = <hl> - токен для выделения частей в тексте;
- separation token = <sep> - токен для разделения сущностей;
- end-of-string token = </s> - токен для обозначения конца строки.

Была использована концепция multitask-learning, в том плане, что модель одновременно обучалась извлекать ответы из текста и генерировать вопросы, но формально перед моделью стояла всего одна задача - минимизация потери при восстановлении target sequence. Гиперпараметры использованные в данной модели: batch_size=8, warmup_steps=0, learning_rate=0.0001, adam_epsilon=1.0e-8, weight_decay=0.01. В результате тренировки получились следующие метрики:

Question answering Exact Match: 75.22;

Question answering F1 Score: 83.8.

3.4 Text complexity prediction

Мой метод предлагает решение этой задачи при помощи блендинга моделей градиентного бустинга, основанных на признаках из компьютерной лингвистики и трансформера RoBERTa base [28]. На данный момент не существует официального датасета касательно этой задачи, поэтому мне пришлось составить собственный.

3.41 Text complexity prediction dataset creation

Одна из известных мне попыток создать датасет текстов с уровнями является OneStopEnglish corpus [38]. Главным недостатком этого датасета является наличие всего 3 уровней языка: C1, B1, A2. Поэтому было принято решение составить собственный датасет, который бы включал все 5 классов уровня CEFR.

Для этого было обработано 7 сайтов агрегаторов для изучения английского языка, полный список можно найти в Приложение 1. В результате было получено 1959 текстов различного уровня: A1=396, A2=458, B1=530, B2=450, C1=125.

3.42 Text complexity prediction model

Для решения этой задачи я использовал модель градиентного бустинга из пакета XGBoost [39] с исходными гиперпараметрами. Данная модель обучалась на признаках, составленных при помощи пакета textstat [40], полный набор признаков можно найти в секции Приложение 1. В результате K-Fold кроссвалидации на 5 частях удалось добиться 0.81 F1 score данной модели.

Также для решения этой задачи была задействована модель distilroberta-base [40] из пакета Huggingface Transformers [34] без каких либо архитектурных изменений. В результате K-Fold кроссвалидации на 5 частях удалось добиться 0.87 F1 score данной модели. Что превосходит предыдущую модель, которая основывалась не на векторных представлениях слов, а не

специальных текстовых статистических признаках.

После этого был произведен блендинг предыдущих моделей в отношении 0.5, в результате чего удалось добиться 0.90 F1 score.

В дополнение к этому был проведен анализ текста на связь признаков из компьютерной лингвистики и уровня сложности согласно CEFR, подробнее об этом можно прочитать в Приложении 3.

4 Заключение, дальнейшая работа

В этой работе были рассмотрены примеры приложения классических задач NLP для решения специфических задач из сферы изучения английского языка. Я считаю, что предложенные здесь методы являются универсальными для большинства романских языков.

Все представленные датасеты нуждаются в существенной доработке и увеличении их объема для получения более качественных результатов. Модели трансформеров, которые использовались мною, требуют много ресурсов для своего запуска. Поэтому стоит подумать об их дистилляции или прунинге. Также их можно использовать для создания датасетов в автоматическом режиме для обучения более простых, но быстрых моделей.

В дальнейшем, планируется попробовать больше подходов для решения задач представленных здесь.

Список литературы:

- [01] - The Global Online Language Learning Market to grow USD 12,452.63 Million by 2025, at a CAGR of 13.18%. [Электронный ресурс] // reports.valuates.com URL: https://reports.valuates.com/market-reports/360I-Auto-1O294/the-global-online-language-learning

- [02] - Education: From disruption to recovery [Электронный ресурс] // en.unesco.org URL: https://en.unesco.org/covid19/educationresponse

- [03] - Annotating tense, mood and voice for English, French and German [Электронный ресурс] // cis.uni-muenchen.de URL: https://www.cis.uni-muenchen.de/~fraser/pubs/ramm_acldemo2017.pdf

- [04] - Sentence tense detector for Fin NLP [Электронный ресурс] // github.com URL: https://github.com/alexcorvi/fin-tense

- [05] - brainly page [Электронный ресурс] // brainly.com URL: https://brainly.com/subject/english

- [06] - ГДЗ по английскому языку за 11 класс [Электронный ресурс] //gdz.ru URL: https://gdz.ru/class-11/english/

- [07] - TOEFL [Электронный ресурс] // toeflgoanywhere.org URL: https://www.toeflgoanywhere.org/

- [08] - IELTS [Электронный ресурс] // ielts.org URL: https://www.ielts.org/

- [09] - duolingo english test [Электронный ресурс] // englishtest.duolingo.com URL: https://englishtest.duolingo.com/home

- [10] - SQuAD The Stanford Question Answering Dataset [Электронный ресурс] // rajpurkar.github.io URL: https://rajpurkar.github.io/SQuAD-explorer/

- [14] - Exploring Transfer Learning with T5: the Text-To-Text Transfer Transformer [Электронный ресурс] // ai.googleblog.com URL: https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html

- [15] - How Much Knowledge Can You Pack Into the Parameters of a Language Model? [Электронный ресурс] // arxiv.org URL: https://arxiv.org/abs/2002.08910

- [16] - Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds [Электронный ресурс] // arxiv.org URL: https://arxiv.org/abs/1911.02365

- [17] - Question Generation for Language Learning:
From ensuring texts are read to supporting learning
[Электронный ресурс] // aclweb.org URL:
https://www.aclweb.org/anthology/W17-5038.pdf
- [18] - ProphetNet: Predicting Future N-gram for Sequence-to-Sequence
Pre-training [Электронный ресурс] // arxiv.org URL:
https://arxiv.org/pdf/2001.04063.pdf
- [19] - Automatically Generating Cause-and-Effect Questions from Passages
[Электронный ресурс] // aclweb.org URL:
https://www.aclweb.org/anthology/2021.bea-1.17.pdf
- [20] - Deep Learning Based Question Generation Using T5 Transformer
[Электронный ресурс] // books.google.ru
https://books.google.ru/books?id=5rUcEAAAQBAJ&lpg=PA243&ots=NYuAt9ap5W&dq=t5%20question%20generation&lr&pg=PA244#v=onepage&q&f=false
- [21] - Neural question generation using transformers [Электронный ресурс] //
github.com URL: https://github.com/patil-suraj/question_generation
- [22] - International language standards [Электронный ресурс] //
cambridgeenglish.org URL:
https://www.cambridgeenglish.org/exams-and-tests/cefr/
- [23] - Reading level assessment using support vector machines and statistical
language models [Электронный ресурс] // dl.acm.org:
https://dl.acm.org/doi/10.3115/1219840.1219905
- [24] - Linguistic Features for Readability Assessment [Электронный ресурс] //
aclweb.org URL: https://www.aclweb.org/anthology/2020.bea-1.1.pdf
- [25] - Introduction to the CoNLL-2003 Shared Task:
Language-Independent Named Entity Recognition
[Электронный ресурс] // aclweb.org URL:
https://www.aclweb.org/anthology/W03-0419.pdf
- [26] - spacy [Электронный ресурс] // spacy.io URL: https://spacy.io/
- [27] - labelbox [Электронный ресурс] // labelbox.com URL:
https://labelbox.com/

- [28] - RoBERTa: A Robustly Optimized BERT Pretraining Approach [Электронный ресурс] // arxiv.org URL: https://arxiv.org/abs/1907.11692

- [29] - A Python framework for sequence labeling evaluation(named-entity recognition, pos tagging) [Электронный ресурс] // github.com URL: https://github.com/chakki-works/seqeval

- [30] - scikit-learn [Электронный ресурс] // scikit-learn.org URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

- [31] - CTRL: A CONDITIONAL TRANSFORMER LANGUAGE MODEL FOR CONTROLLABLE GENERATION [Электронный ресурс] // arxiv.org URL: https://arxiv.org/pdf/1909.05858.pdf

- [32] - medium [Электронный ресурс] // medium.com URL: https://medium.com/

- [33] - Language Models are Unsupervised Multitask Learners [Электронный ресурс] // cdn.openai.com URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

- [34] - OpenAI GPT2 [Электронный ресурс] // huggingface.co URL: https://huggingface.co/transformers/model_doc/gpt2.html

- [35] - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer [Электронный ресурс] // arxiv.org URL: https://arxiv.org/pdf/1910.10683.pdf

- [36] - T5 [Электронный ресурс] // huggingface.co URL: https://huggingface.co/transformers/model_doc/t5.html

- [37] - Exploratory data analysis [Электронный ресурс] // en.wikipedia.org URL: https://en.wikipedia.org/wiki/Exploratory_data_analysis

- [38] - OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification [Электронный ресурс] // aclweb.org URL: https://www.aclweb.org/anthology/W18-0535.pdf

- [39] - xgboost [Электронный ресурс] // xgboost.readthedocs.io URL: https://xgboost.readthedocs.io/en/latest/

- [40] - Oxford 3000 and 5000 The most important words to learn in English. [Электронный ресурс] // oxfordlearnersdictionaries.com URL: https://www.oxfordlearnersdictionaries.com/wordlists/oxford3000-5000

- [41] - Hierarchical Neural Story Generation [Электронный ресурс] // arxiv.org URL: https://arxiv.org/abs/1805.04833

Приложение 1

Полный список тем текстов взятых с сайта Medium:

business, ux, design, economy, freelancing, leadership, marketing, startups, artificial-intelligence, data-science, neuroscience, productivity, gaming, social-media, fiction, science, education, travel, mindfulness, mental-health

Полный список признаков использованных для предсказания уровня текста:

Syllable Count, Lexicon Count, Sentence Count, The Flesch Reading Ease formula, The Flesch-Kincaid Grade Level, The Fog Scale (Gunning FOG Formula), The SMOG Index, Automated Readability Index, The Coleman-Liau Index, Linsear Write Formula, Dale-Chall Readability Score.

Полный список сайтов обработанных для задачи [3.42]:

breakingnewsenglish.com, continuingstudies.uvic.ca, englishteststore.net, esl-lounge.com, learnamericanenglishonline.com, learnenglish.britishcouncil.org, lingua.com

Приложение 2, Controllable question generation with grammar constructions

Благодаря модели, полученной в секции [3.13], удалось в автоматическом режиме разметить вопросы из датасета SQuAD 1.1 аналогично тому, как это было сделано в секции [3.21]. В результате тренировки была получена модель, которая способна изменять концентрацию грамматических конструкций в целевом вопросе. Во время ручной проверки было замечено, что модель верно определяет направление генерации, но по большей части все вопросы оказываются грамматически неверными. Пример подобной генерации представлен ниже:

Original question: What is the non-compulsory educational level that follows higher education?

a1_have_has_got: What type of higher education has a non-compulsory level?

a1_past_simple_reg: What type of higher education followed?

a1_be_have_do_in_the_past: What type of higher education had a non-compulsory level?

a1_can: What type of higher education can follow?

a1_comparative_short: What is the most common type of higher education?

a1_possesive_s_sing: What is higher education's non-compulsory level?

a1_there_was_were: There was higher education that follows what?

a1_there_will_be: There will be higher education followed by what type of school?

Приложение 3, Исследование зависимости статистических признаков и сложности текста

В результате анализа текстов при помощи модели, представленной в секции [3.13], пакета textstat и признаков созданных на основе списка уровня слов согласно словарю Oxford [40]. В следствие чего было выяснено, что наибольшее влияние играют следующие признаки:

- Fórmula de Crawford

- difficult words(данный список слов представлен самой библиотекой)

- The SMOG Index

- Word Counts B2 (количество слов в тексте, которые относятся к уровню B2 согласно словарю Oxford)

Также было выяснено, что количество грамматических конструкций, представленных в секции [3.12], имеют слабую корреляцию с уровнем сложности текста согласно CEFR.