Оглавление

Введение

Целью упрощения текста является его преобразование в более легкую для чтения и понимания форму. Решение этой задачи всегда было необходимо для отдельных групп людей[1][2] и для подготовки данных в других задачах обработки естественного языка[3], а ее актуальность возрастает в связи с резким увеличением количества неструктурированных текстовых данных из-за развития Интернета.

В процессе упрощения текста нужно рассматривать сразу несколько его сотставляющих. Например, необходимо делать проще его лексику и структуру, при этом сохраняя смысловую часть неизменной.

Оценка качества упрощения также не является тривиальной задачей. Из-за субъективности такой оценки, несмотря на появление метрик, которые позволяют это делать автоматически, оценка текста несколькими людьми все еще считается наиболее достоверной[4].

Целью данной работы является выбор метода, который наиболее полно решает задачу упрощения текстов.

В рамках выполнения работы необходимо решить следующие задачи:

- провести анализ предметной области;
- рассмотреть существующие метрики оценки качества упрощения;
- провести анализ существующих решений задачи упрощения текстов;
- сформулировать критерии выбора решения;
- на основе этих критерив провести классификацию решений;
- определить, какой метод или методы являются лучшими по совокупности критериев.

1 Анализ предметной области

В данном разделе вводятся основные определения и описываются важность и актуальность задачи упрощения текстов.

1.1 Задача упрощения текстов

Существуют различные формулировки задачи упрощения текста. Так, в статье [5] даются определения в двух смыслах:

- упрощение текста в узком смысле - это процесс уменьшения его лингвистической сложности при сохранении исходной информации и смысла;
- в более широком смысле упрощение текста охватывает и другие операции: смысловое изменение для упрощения как формы, так и содержания; краткое изложение текста для исключения второстепенной или избыточной информации.

В статье [6] упрощением предложений называют процесс, целью которого является получение более легкого для чтения и понимания текста за счет уменьшения его лексической и структурной сложности.

Лингвистическая сложность обычно рассчитывается на основе трех факторов - количественных параметров текста (например, длина и количество слов и предложений), его качественных параметров (например, структурированность текста, насколько сложными с точки зрения грамматики являются используемые в нем предложения, содержится ли в нем специфическая терминология), а также на уровне подготовки читателей[7]. При этом под лексической состаляющей сложности подразумевают сложность текста с точки зрения используемых в нем слов (их длина или частотность употребления), а под структурной - с точки зрения сложности грамматики его предложений (количество простых предложений в составе сложного, наличие сравнительных, причастных и деепричастных оборотов и т. д.)[8].

При этом задача упрощения текстов относится к области NLP (Natural language processing, обработка текстов на естественном языке) и имеет много общего с другими задачами из этой сферы - машинным переводом, перефразированием и обобщением (резюмированием) текста[9].

Важно отметить отличие упрощения текста от его обобщения, так как эти задачи зачастую путают. Отличие заключается в том, что во втором случае основное внимание уделяется сокращению длины исходных данных и удалению из них второстепенной информации. И хотя обобщенные тексты, как правило, короче, это не всегда так, и обобщение может привести к увеличению длины полученных предложений[4], что сделает текст более сложным для чтения. В рамках же упрощения текста основной целью считается сделать его более легким для восприятия, а сокращение смыслового содержания рассматривается только при использовании широкого понятия термина «упрощение».

1.2 Актуальность

В последние десятилетия количество неструктурированных текстовых данных резко возросло в связи с развитием Интернета. Как следствие, возросла и потребность в их упрощении, что показано на рисунке 1.1.

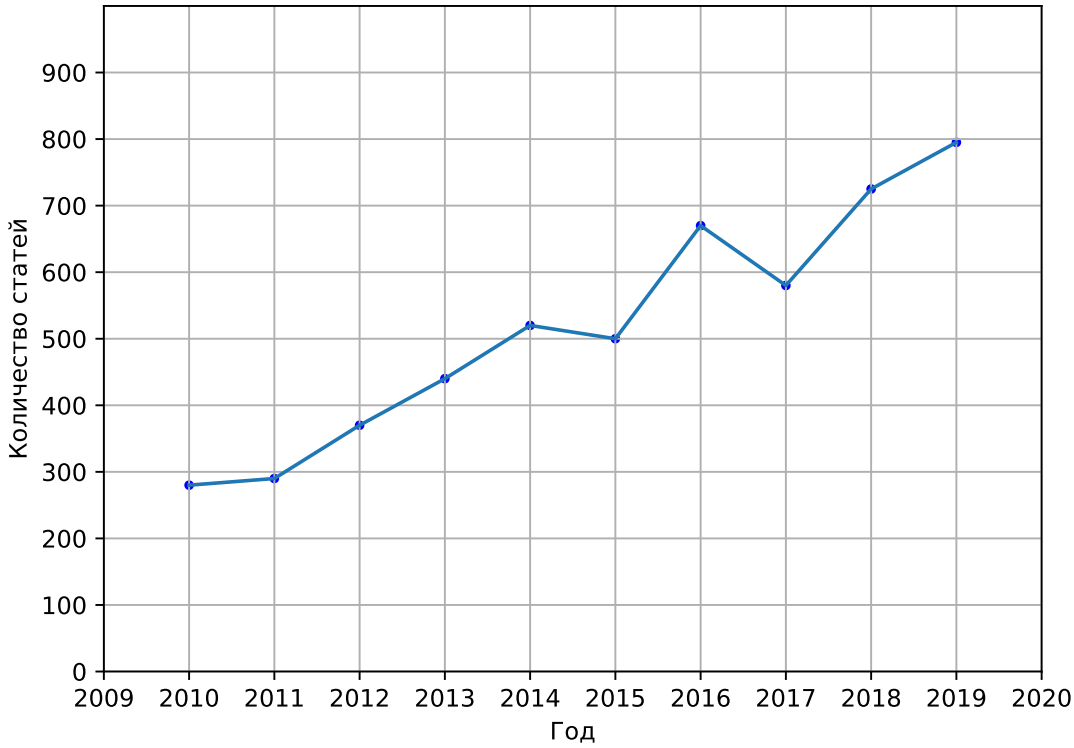

Рисунок 1.1 – График роста интереса к теме упрощения текста. Создано на основе статистики Google Scholar по поисковым запросам «Text Simplificatio» (упрощениие текстов), «Lexical Simplification» (лексическое упрощение), «Syntactic Simplification» (синтаксическое упрощение)[10]

Упрощение текстов необходимо для различных задач и целевых аудиторий:

- в качестве этапа подготовки текста перед его обобщением [3];
- для людей, изучающих иностранный язык, и детей, учащихся читать (требуется лексическое упрощение для сокращения количества специализированных и нечастотных слов)[1];
- для людей с дислексией и афазией, для которых длинные слова и предложения могут представлять трудности;
- для людей, страдающих аутизмом (необходимо уменьшать количество образных выражений и синтаксическую сложность)[2].

1.3 Данные

Упрощение предложений привлекло многих исследователей в связи с появлением больших параллельных корпусов. Наиболее известные из них составлены из английских текстов, например, PWKP и Wiki-large. Последний представляет собой большой параллельный корпус на английском языке, состоящий из сложных предложений, взятых из Википедии, и их выровненных упрощенных версий.

Подобных данных на других языках значительно меньше. Большой корпус на русском языке был собран, когда задача упрощения текстов была предложена в рамках международной конференции по компьютерной лингвистике и интеллектуальным технологиям DIALOGUE 2021[1]. Организаторы подготовили обучающие и тестовые наборы на русском языке с использованием краудсорсинговой платформы, а также перевели тексты из Википедии (корпус RuWikiSimple). Другим источником данных могут стать результаты перевода, выполненные при помощи перефразирования[11].

Основная проблема в упомянутых данных заключается в том, что имеет место фокусировка на текстах Википедии. Это ограничивает исследования и приводит к неадекватности моделей на других типах данных [11].

Выводы из анализа предметной области

Таким образом, акутальность задачи упрощения тексов в последнее десятилетие увеличивается, формируются новые корпуса на различных языках для обучения моделей. При этом поставленная задача схожа с другими задачами из сферы NLP, а если использовать ее более широкое понятие, то она будет трудно отличима от задачи обобщения. Поэтому, чтобы разграничить эти два понятия, в данной работе будет рассматриваться задача упрощения предложений в более узком смысле: как процесс, целью которого является получение более легкого для чтения и понимания текста за счет уменьшения его лексической и структурной сложности[6].

[1] http://www.dialog-21.ru/dialogue2021/results/

2 Классификация существующих решений

В данном разделе описываются существующие метрики качества упрощенного текста, предлагаются критерии оценки методов упрощения. Затем описываются известные решения поставленной задачи и приводится их классификация по сформулированным критериям.

2.1 Метрики

Оценка качества упрощения текста - довольно субъективная задача, трудно переводимая на язык компьютера. Поэтому до сих пор предпочтение отдается человеку, который оценивает упрощенное предложение с точки зрения его грамматической корректности, сохранения смысла и простоты, используя шкалы Лайкерта[1]. Однако тенденция к использованию технологии обучения без учителя потребовала адаптации уже существующих показателей для автоматической оценки качества упрощения или разработки новых метрик.

2.1.1 Индексы удобочитаемости

Индексы удобочитаемости используются в США для присвоения любому тексту уровня его «простоты». Обычно для такой оценки применяется сразу несколько формул, которые учитывают количество предложений, слогов, общее количество слов и количество редких слов в рассматриваемом тексте. Отличаются эти формулы лишь коэффициентами, расположением членов в формуле или способом интерпретации результата.

Например, в тесте Флэша-Кинкайда предполагается, что чем меньше слов в предложениях и чем короче слова, тем более простым является текст. В результате применения формулы 2.1 корпус получает оценку, по которой с по-

[1]Шкала Лайкерта - это ординальная (порядковая) шкала ответов на вопрос или утверждений, расположенных в иерархической последовательности, например, от «полностью согласен» через «затрудняюсь ответить» и до «категорически не согласен»[12].

мощью специальной таблицы интерпретируется уровень образования, необходимый для понимания текста[13].

$$206.835 - 1.015 \cdot \left(\frac{\text{количество слов}}{\text{количество предложений}}\right) - \\ -84.6 \cdot \left(\frac{\text{количество слогов}}{\text{количество слов}}\right) \quad (2.1)$$

Основной недостаток индексов удобочитаемости заключается в том, что используемые в них формулы «поощряют» короткие предложения с простыми словами, но не способны распознавать грамматически неправильные результаты или упрощенные предложения, некорректно передающие смысл исходного.

2.1.2 SARI

Метрика SARI (**s**ystem output **a**gainst **r**eferences and against the **i**nput sentence - результат системы против эталонного решения и исходного предложения) была разработана в 2016 году специально для задачи упрошения и в настоящее время считается наиболее применимой[14].

Авторы SARI заметили, что, в отличие от метрик качества машинного перевода, где исходное предложение записано на другом языке, метрики качества упрощения могут использовать еще один источник для оценки результирующего предложения - само исходное предложение, так как оно записано на том же языке, что и упрощенное. Разработанная ими метрика напрямую оценивает, насколько корректно были выбраны слова, которые были добавлены, удалены или же сохранены моделью.

Для вычисления значения SARI используются исходное сложное предложение, несколько примеров его упрощения, предложенные, как правило, людьми, и результат, полученный моделью. Каждое предложение рассамтривается как множество n-грамм - последовательностей из n слов.

SARI поощряет добавление моделью n-граммы g в упрощенное предложение O, если ее не было в исходном предложении I, но она также была добавлена в любом из примеров упрощения R, то есть если $g \in O \cap \bar{I} \cap R$.

Точность (precision) $p_{add}(n)$ и полнота (recall) $r_{add}(n)$ операции добавления n-грамм в предложение рассчитывается, соответственно, по формулам 2.2 и 2.3:

$$p_{add}(n) = \frac{\sum_{g \in O} min\left(\#_g\left(O \cap \bar{I}\right), \#_g\left(R\right)\right)}{\sum_{g \in O} \#_g\left(O \cap \bar{I}\right)}, \tag{2.2}$$

$$r_{add}(n) = \frac{\sum_{g \in O} min\left(\#_g\left(O \cap \bar{I}\right), \#_g\left(R\right)\right)}{\sum_{g \in O} \#_g\left(R \cap \bar{I}\right)}, \tag{2.3}$$

где $\#_g(\cdot)$ - это бинарный индикатор появления n-граммы g в данном множестве (и долевой индикатор в некоторых следующих формулах) и

$$\#_g\left(O \cap \bar{I}\right) = max\left(\#_g\left(O\right) - \#_g\left(I\right), 0\right), \tag{2.4}$$

$$\#_g\left(R \cap \bar{I}\right) = max\left(\#_g\left(R\right) - \#_g\left(I\right), 0\right). \tag{2.5}$$

Так, в приведенном ниже примере добавление униграммы «заранее» в первом и втором результатах поощряется как в $p_{add}(n)$, так и в $r_{add}(n)$, в то время как добавление «были» в первом результате штрафуется в $p_{add}(n)$.

Таблица 2.1 – Сравнение упрощений системой с примерами

Ввод	Около 5 способов предварительно апробированы
Пример 1	Около 5 способов предварительно проверены
Пример 2	Около 5 способов заранее апробированы
Пример 3	5 способов заранее апробированы
Результат 1	Около 5 способов были заранее испытаны
Результат 2	Около 5 способов заранее опробованы
Результат 3	Около 5 способов предварительно опробованы

SARI поощряет сохранение n-граммы моделью, если она была также сохранена в примерах. При этом учитывается количество примеров, в которых эта n-грамма была сохранена, для чего вводится R'. Например, униграмма «около» в приведенном выше примере встречается в двух из общего числа

$r = 3$ примеров, поэтому ее вклад в точность и полноту операции сохранения взвешивается на $\frac{2}{r} = \frac{2}{3}$. Точность $p_{keep}(n)$ и полнота $r_{keep}(n)$ операции сохранения n-грамм в предложении рассчитывается, соответственно, по формулам 2.6 и 2.7:

$$p_{keep}(n) = \frac{\sum\limits_{g \in I} min\left(\#_g\left(I \cap O\right), \#_g\left(I \cap R'\right)\right)}{\sum\limits_{g \in I} \#_g\left(I \cap O\right)}, \tag{2.6}$$

$$r_{keep}(n) = \frac{\sum\limits_{g \in I} min\left(\#_g\left(I \cap O\right), \#_g\left(I \cap R'\right)\right)}{\sum\limits_{g \in I} \#_g\left(I \cap R'\right)}, \tag{2.7}$$

где

$$\#_g\left(I \cap O\right) = min\left(\#_g\left(I\right), \#_g\left(O\right)\right), \tag{2.8}$$

$$\#_g\left(I \cap R'\right) = min\left(\#_g\left(I\right), \frac{\#_g\left(R\right)}{r}\right). \tag{2.9}$$

Для операции удаления рассчитывается только точность, так как удаление «лишних» n-грамм значительно хуже для качества полученного результата, чем не удаление нужных n-грамм. Здесь также вводится множество $\bar{R}'$ для взвешивания n-грамм. Точность $p_{del}(n)$ операции удаления n-грамм из предложения рассчитывается по формуле 2.10:

$$p_{del}(n) = \frac{\sum\limits_{g \in I} min\left(\#_g\left(I \cap \bar{O}\right), \#_g\left(I \cap \bar{R}'\right)\right)}{\sum\limits_{g \in I} \#_g\left(I \cap \bar{O}\right)} \tag{2.10}$$

где

$$\#_g\left(I \cap \bar{O}\right) = max\left(\#_g\left(I\right) - \#_g\left(O\right), 0\right), \tag{2.11}$$

$$\#_g\left(I \cap \bar{R}'\right) = max\left(\#_g\left(I\right) - \frac{\#_g\left(R\right)}{r}, 0\right). \tag{2.12}$$

При этом достаточность удалений отражается в точности операции сохранения. В итоге для расчета SARI для одного упрощенного предложения используется среднее арифметическое точности $P_{operation}$ и полноты $R_{operation}$

операций добавления, сохранения и удаления по всем n-граммам предложения:

$$SARI = d_1 F_{add} + d_2 F_{keep} + d_3 P_{del}, \tag{2.13}$$

где $d1 = d2 = d3 = \frac{1}{3}$ и

$$P_{operation} = \frac{1}{k} \sum_{n=1}^{k} p_{operation}(n), \tag{2.14}$$

$$R_{operation} = \frac{1}{k} \sum_{n=1}^{k} r_{operation}(n), \tag{2.15}$$

$$F_{operation} = \frac{2 P_{operation} R_{operation}}{P_{operation} + R_{operation}}, \tag{2.16}$$
$$operation \in \{del, keep, add\},$$

где k - наибольший порядок n-грамм (авторы SARI используют k=4).

Результаты исследования авторов показали высокую корреляцию оценки метрикой SARI с оценками качества упрощения человеком, однако, эта функция требует нескольких «справочных» примеров упрощения, которые не всегда доступны. Более того, она более сфокусирована на оценке упрощения на уровне лексики, и слабо способна распознать степень того, насколько проще стало предложение на уровне его структуры.

2.1.3 SAMSA

Метрика SAMSA (**s**implification **a**utomatic evaluation **m**easure through **s**emantic **a**nnotation, (автоматическая оценка упрощения с помощью семантической аннотации), разработанная в 2018 году, была призвана исправить последний недостаток SARI, то есть оценивать как качество лексического упрощения, так и простоту полученного предложенияя с точки зрения его структуры[15].

SAMSA предполагает, что оптимальное разделение исходного предложения - это такое, при котором каждой возможной структуре подлежащее-сказуемое присваивается собственное предложение, и измеряет, в какой степени это утверждение справедливо для рассматриваемой пары ввода-вывода,

используя семантическую структуру. Например, одно сложно предложение «Тюльпан, сорванный Петей, не понравился Маше, потому что она любит розы.» следует разделить на три простых: «Петя сорвал тюльпан. Тюльпан не понравился Маше. Потому что она любит розы.»

Эта метрика фокусируется на основных семантических компонентах предложения и менее внимательно относится к удалению других структурных единиц.

Авторы SAMSA также демонстрируют высокую корреляцию оценки этой метрикой с оценками качества упрощения человеком. Но так как SAMSA появилась относительно недавно, она в настоящее время еще широко не используется. Более того, она не решила другую проблему SARI - потребность в «справочных» примерах упрощения.

2.2 Критерии оценки решений

Как уже упоминалось, данная работа фокусируется на поиске решения задачи упрощения в узком смысле, как на задаче получения более легкого для чтения и понимания текста за счет уменьшения его лексической и структурной сложности[6]. Поэтому в первую очередь рассматриваемые решения будут классифицироваться на те, что решают задачу упрощения в широком смылсе и те, что решают ее в узком смысле.

Из данного выше определения следует вывод, что упрощение можно условно разделить на лексическую и структурную составляющую. Поэтому другими критериями оценки могут стать ответы на вопросы, учитывает ли рассматриваемое решение каждую из составляющих.

Целью такой классификации будет поиск решения, которое работает с задачей упрощения текстов в узком смысле и при этом учитывает как лексическое, так и структурное упрощение предложений текста.

Ранее было приведено несколько метрик, которые способны автоматически оценивать качество упрощения. Однако решения, которые будут рассмотрены далее, обучались и тестировались на разных корпусах, оценивались разными метриками, то есть нет готовых данных для объективного сравнения решений по определенной метрике. В связи с ограниченным временем на вы-

полнение работы, нет возможности самостоятельно получить эти данные. Поэтому классификация решений по их эффективности проводиться не будет.

2.3 Подходы к решению задачи

Глобально выделяют два подхода к решению задачи упрощения текстов - экстрактивный (извлекающий) и абстрактный.

Большинство ранних работ, посвященных задаче упрощения текстов, использовали экстрактивный подход - выделение в документе тех предложений, которые передают больше информации. Полученные тексты будут легче для чтения и восприятия, так как они становятся значительно короче исходных, а многие редко употребляемые слова будут отброшены, так как не попадут в выбранные предложения.

Этот подход достаточно прост в реализации, однако он подходит лишь для решения задачи упрощения текстов в широком смысле, так как из-за удаления предложений, несущих малое количество информации, исходный смысл корпуса будет передан не в полной мере. Поэтому решения экстрактивного подхода в дальнейшем в данной работе рассматриваться не будут.

С ростом доступности вычислительных ресурсов и количества исследований в области NLP для решения задачи упрощения текстов все чаще стал использоваться абстрактный подход, подразумевающий **переписывание** передаваемого текста предложение за предложением[16].

2.4 Решения абстрактного подхода

Решения задачи упрощения текстов, относяшиеся к абстрактному подходу, можно разделить на два класса: те, что основаны на текстовых заменах, и те, что подразумевают генерацию нового текста.

2.4.1 Текстовые замены

Текстовые замены на уровне слов или целых фраз лежали в основах первых работ, посвященных абстрактному подходу к упрощению текстов [17]. Этот процесс фокусируется на сокращении лексического содержания текста, но не принимает во внимание такие подзадачи упрощения структурной составляющей, как грамматическое или синтаксическое упрощение[4]. Таким образом, эти решения не учитывают все аспекты решаемой задачи, и поэтому в дальнейшем в работе рассматриваться не будут.

2.4.2 Генерация нового текста

Современные решения абстрактного подхода, включают в себя разбиение сложных предложений на более простые, удаление редко употребляемых слов и генерацию на этой основе нового текста. Это стало возможным благодаря появлению нейронных сетей, в частности, рекуррентных нейронных сетей (RNNs)[2], которые позволяют решать задачи «от последовательности к последовательности» (seq2seq)[3].

Последовательность действий в решении задач, сведенных к seq2seq, является универсальной. Предварительная обработка данных включает очистку входного и результирующего текста, удаление знаков препинания и специальных символов, формирование словаря типичных слов. Далее входные и результирующие предложения преобразуются в числовую форму с вектором одинаковой длины либо путем их усечения, либо путем их дополнения. Затем модель обучается, после чего новые поступающие данные также сначала векторизуются, преобразуются к определенной длине и упрощаются, а только после этого преобразовываются обратно из числовой формы в текстовую.

Далее будут рассмотрены различные виды решений задачи упрощения текста, относящиеся к классу тех, которые используют генерацию нового тек-

[2]Рекуррентная нейронная сеть, или RNN, - это сеть, которая работает с последовательностью и использут собственные промежуточные выходные данные в качестве входных данных для последующих шагов [18]

[3]Сеть sequence-to-sequence («от последовательности к последовательсности»), или сеть seq2seq, или сеть кодировщика-декодера, представляет собой модель, состоящую из двух RNN, называемых кодировщиком и декодером. Кодер считывает входную последовательность и выдает один вектор, а декодер считывает этот вектор для создания выходной последовательности [18]

ста.

Синтаксическое упрощение

Цель синтаксического упрощения заключается в выявлении грамматически сложных частей текста и их переписывание для облегчения понимания. Такое упрощение может включать разделение длинных предложений на более короткие фрагменты, переписывание предложений со страдательным залогом так, чтобы в них использовался залог действительный[4], разрешение двусмысленностей и анафор[4]. По ходу выполнения синтаксического упрощения удается заменять слова, которые считаеются «сложными» из-за наложения морфем, участвующих в переводе слова из одной части речи в другую, на их более простые, оригинальные версии.

Основополагающей работой в области синтаксического упрощения была система автоматического создания правил переписывания предложений[20], которая брала аннотированные корпуса и изучала возможные принципы упрощения для конкретной предметной области.

Более поздние работы по синтаксическому упрощению были сосредоточены на улучшении структуры выходного текста - обеспечении того, чтобы предложения появлялись в правильном порядке [21]. Также этот подход стали использовать для распознавания именованных сущностей (Named Entity Recognition, NER), особенно в области медицины [22].

Синтаксическое упрощение обычно выполняется в три этапа. Пример применения каждого этапа к предложению показан на рисунке 2.1

[4] Действительный залог имеют глаголы переходные, обозначающие действие, производимое субъектом и активно направленное на объект. Действительный залог имеет синтаксическую характеристику: субъект действия является подлежащим, а объект - дополнением в винительном падеже без предлога: Мир победит войну.

Страдательный залог выражается присоединением к глаголам действительного залога аффикса -ся (ср.: Рабочие строят дома. - Дома строятся рабочими). Кроме того, значение страдательного залога может быть выражено формами страдательных причастий - полных и кратких. Например: Мать любима (любимая). Тема изучена (изученная)[19].

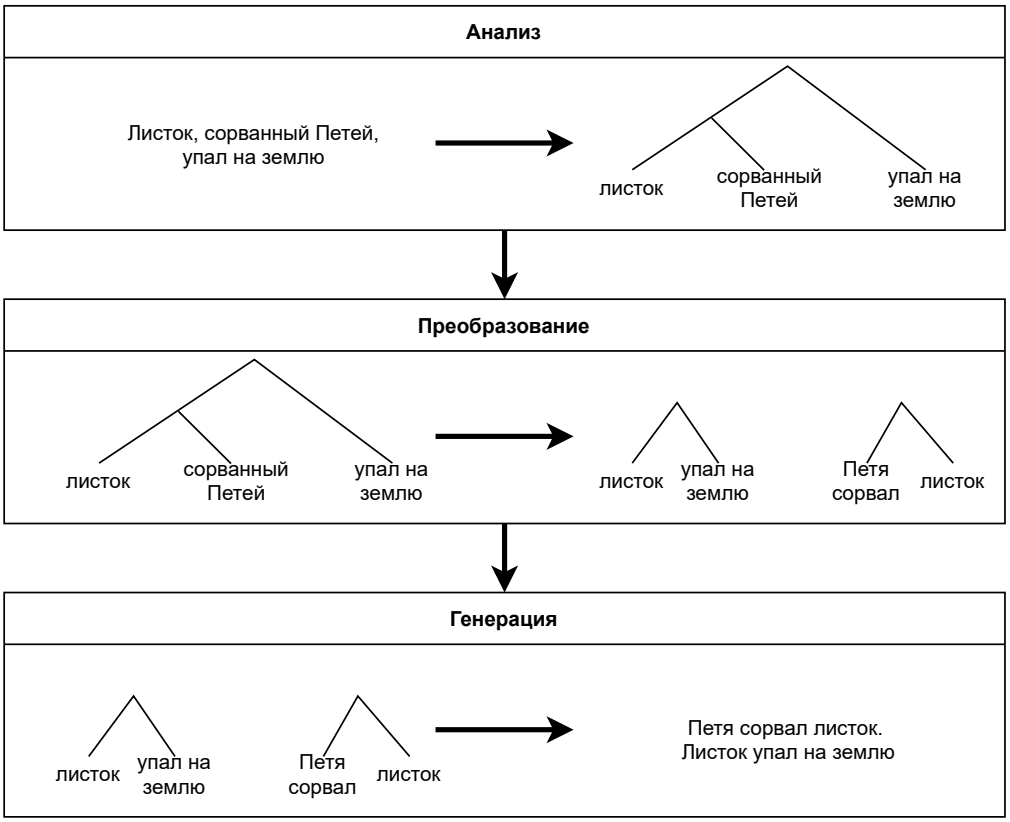

Рисунок 2.1 – Три этапа синтаксического упрощения

1. Анализ. Определяется структура предложения и создается дерево синтаксического анализа. Это может быть сделано на разных уровнях детализации, но наилучшие результаты достигаются на довольно высоком уровне, когда слова и фразы группируются в так называемые «супертеги», представляющие собой фрагменты исходного предложения. Такие теги могут быть объединены по обычным грамматическим правилам и являются структурированной версией текста. На этом же этапе простой проверкой по заранее заданным правилам или с помощью бинарного классификатора SVM (метод опорных векторов)[4] определяется сложность предложения.

2. Преобразование. Дерево синтаксического анализа модифицируется в соответствии с набором правил, которые выполняют операции упрощения. Например, разбиение сложного предложения на несколько более простых, перестановка или удаление полученных предложений[23].

3. Генерация. В текст вносятся дополнительные изменения для улучшения согласованности и удобства чтения, добавляются знаки препинания.

Синтаксическое упрощение считалось важным компонентом систем упрощения текстов и было реализовано в системах, которые повсеместно используются как вспомогательные, например, в PSET[24] и Porsimples[25].

Преимущества синтаксического упрощения заключаются в его высокой точности и применимости к другим задачам NLP [4].

Недостатком является трудоемкость создания и проверки применимости правил перезаписи. Но в последнее время достижения в области методов глубокого обучения привели к автоматизации процесса обнаружения возможности применения синтаксического упрощения.

Еще один недостаток - лексическое упрощение выполняется не в полной мере. По ходу выполнения синтаксического упрощения удается заменять лишь некоторые слова, которые были упомянуты выше, но не редкоупотребляемые и специализированные слова.

Статистический машинный перевод

Автоматизированный машинный перевод является устоявшейся техникой в NLP. Эта задача подразумевает автоматическое преобразование лексики и синтаксиса одного языка в правила другого, в результате чего получается переведенный текст. Машинный перевод был успешно применен[4] к задаче упрощения текстов путем ее переформулирования в задачу «одноязычного перевода». То есть задача упрощения была сведена к переводу с исходного «сложного» языка на целевой «простой» язык.

Разновидностью машинного перевода является статистический машинный перевод (Statistical Machine Translation (SMT)), основанный на статистических моделях, которые изначально «ничего не знают» о правилах и лингвистике, а затем изучают большие объемы пар предложений из выровненных двуязычных корпусов, настраивая свои параметры (наиболее вероятный вариант перевода того или иного слова), а затем применяются к новым текстам.

Например, модель для перевода с русского на английский изучила перевод одного предложения: «Я вижу дом» в «I see a house». Если теперь запросить у нее перевод слова «дом», она предположит, что слово с равной вероятностью переводится как «I», «see», «a» или «house». Но если предоставить модели еще одно соответствие, что предложение «Этот дом большой» переводится как «That house is big», то из анализа уже двух сопоставлений переводчик

отметит, что в переводах все слова встретились по одному разу, а «house» — дважды, равно как и слово «дом» (и никакое другое) в исходных предложениях. А значит, по сравнению со всеми остальными вариантами увеличивается вероятность соответствия «дом = house» и между ними установилась связь.

Задача перевода облегчается, когда исходный и целевой языки схожи, и для преобразования предложения требуется минимальное число изменений его структуры. И именно этот тип машинного перевода был применен к задаче упрощения текстов[4].

Эффективность применения статистического машинного перевода для задачи упрощения предложений в значительной степени зависит от набора данных, используемых для обучения модели. Например, если в них содержатся слишком длинные исходные предложения или упрощеннные предложения слишком сильно оличаются по своей структуре от входных, то этот подход не позволит отслеживать и верно сопоставлять различные части предложений. Кроме того, данный метод не учитывает знаки препинания и разбиение сложных предложений, что зачастую приводит к потере контекста и получению структурно сложных предложений.

Использование методов глубокого обучения

Глубокое обучение - это разновидность машинного обучения, в которой нейронные сети и алгоритмы, изначально являвшиеся попытками моделирования структуры и функционирования человеческого мозга, обучаются на большом объеме данных для создания шаблонов принятия решений. Этот подход позволяет обучаться путем многократного выполнения задач и настройки модели для улучшения результата[26].

Методы глубокого обучения после своего появления стали активно применяться для решения задачи упрощения текстов, сформулированной в терминах моделирования seq2seq. Однако в ранних моделях seq2seq были две существенные проблемы.

Во-первых, это неточность результата. Эффективность моделей кодировщика-декодера сильно зависит от расположения слов в исходном предложени, поэтому модель зачастую не может расположить редко употребляемые слова в корректную позицию выходного предложения. Один из вариантов решения этой проблемы - добавление так называемых «указателей» на подобные слова

в исходном тексте[27]. Это решение показало многообещающие результаты в сохранении корректного смысла сгенерированного упрощенного текста.

Во-вторых, это повторения в выводе. Данная проблема часто возникает в простых моделях seq2seq из-за того, что так называемые «стоп-слова» («как», «и», «а», «то» и т. д.) встречаются в тексте намного чаще остальных, и модель учится чаще предсказывать их. В частности, именно эта проблема стала основным недостатком в решении, описанном ранее[27]. Для борьбы с этим недостатком было предложено штрафовать модель за повторения с помощью введение векторов «покрытия», «внимания» и «контекста». Эти вектора отслеживают слова, которые перешли из исходного предложения в упрощенное: вектор «внимания» - слова, несущие основной смысл, «контекста» - сопуствующую информацию, «покрытия» - общий переданный объем слов. Именно вектор покрытия призван дополнительно контроллировать вектора «внимания» и «контекста» и штрафовать модель при их наложении друг на друга[16].

Также к задаче упрощенияя текстов было успешно применено глубокое обучение, дополненное обучением с подкреплением[28]. Была разработана модель кодировщика-декодера в сочетании с системой глубокого обучения с подкреплением DRESS (Deep Reinforcement Sentence Simplification, глубокое упрощение предложений с подкреплением), стремящаяся оптимизировать функцию потерь, которая поощряет простые, легко читаемые и сохраняющие исходный смысл результаты упрощения. Обучение с подкреплением позволяет генерировать сразу несколько вариантов результата, а затем выбирать из них наиболее удачный. С помощью этой модели было показано, что такое сочетание дает возможность для предоставления дополнительной (предварительной) информации в данные.

Основная проблема решений, использующих глубокое обучение, состоит в их большой ресурсоемкости, что ограничивает количество параметров в используемых в них моделях. Но появление языковой модели BERT[29] (Bidirectional Encoder Representations from Transformers, двунаправленные представления кодировщика трансформатора), основанной на архитектуре seq2seq и предназначенной для предобучения языковых представлений с целью их последующего применения в широком спектре задач NLP, стало основой многих новых исследований.

Так, в рамках упомянутой в предыдущем разделе конференции DIALOGUE 2021 лучший результат показала модель mBART - многоязыковая версия BART, основанная на архитектуре BERT. Эта архитектура отказывается от использования RNN, что значительно повышает ее эффективность и позволяет обучать более глубокие модели с большим количеством параметров.

Также по результатам конференции было высказано и экспериментально доказано предположение, что выбор между различными моделями, использующими глубокое обучение, оказывает ограниченное влияние на конечный результат. Использование же дополнительных показателей для фильтрации обучающих данных или для выбора наиболее подходящего упрощения из созданных, представляется крайне важным для дальнейшего повышения качества результата.

2.5 Классификация

В таблице 2.2 приведена классификация рассмотренных решений по ранее сформулированным критериям. Прочерки в таблице означают, что решения соответсвующей группы не были рассмотрены в данной работе по отдельности. Записи «шир.» и «узк.» в столбце «Понимание задачи» означают, соответсвенно, что решение рабоает с широким и узким определением задачи упрощения текстов.

Таблица 2.2 – Классификация решений задачи упрощения текстов

Подход	Класс	Решение	Понимание задачи	Учет лексического упрощения	Учет структурного упрощения
Экстрактивный	-	-	Шир.	-	-
Абстрактный	Текстовые замены	-	Узк.	Да	Нет
	Генерация нового текста	Синтаксическое упрощение	Узк.	Частично	Да
		Статистический машинный перевод	Узк.	Да	Нет
		Глубокое обучение	Узк.	Да	Да

Из таблицы видно, что наиболее полным образом задачу упрощения текстов в узком смысле решают методы, использующие глубокое обучение. Как упоминалось выше, в результате эксперимента[11] было показано, что выбор между различными моделями этого метода оказывает меньшее влияние на конечный результат, чем использование дополнительных показателей для фильтрации обучающих данных или для выбора наиболее подходящего упрощения из созданных. В связи с этим было решено провести краткий обзор возможных подходов к выбору таких дополнительных показателей.

2.6 Дополнительные показатели

В рамках соревнования по упрощению текстов, проведенного на конференции DIALOGUE 2021, решения участников сравнивались по метрике SARI. При этом все модели, занявшие первые строки таблицы лидеров, использовали методы глубоко обучения, и большинство - дополнительные метрики для фильтрации тренировочных данных и выбора лучшего кандидата из предложенных моделью вариантов. Авторы двух решений опубликовали статьи, в

которых описали, какие именно дополнительные параметры они использовали.

Второе место заняло решение[30], в котором фильтрация тренировочных данных и выбор лучшего кандидата из предложенных моделью вариантов проводились с помощью разработанной авторами функции $RuSimScore(c, s)$. Она позволяет оценить качество упрощенного предложения s, полученного по исходному предложению c. Эта метрика рассчитывается как произведение четырех функций оценки «простоты» (лексической сложности (LS), глубины дерева зависимостей (DD), длины предложения (LeS), легкости чтения (RS)), и двух функций оценки сохранения содержания (косинусного сходства ($SimS$) и сохранения именованных сущностей (NS)):

$$RuSimScore(c, s) = LS^{\alpha}(c, s) \cdot DD^{\beta}(c, s) \cdot LeS^{\gamma}(c, s) \cdot \\ \cdot RS^{\delta}(c, s) \cdot SimS^{\varepsilon}(c, s) \cdot NS^{\zeta}(c, s), \tag{2.17}$$

где $\alpha, \beta, \gamma, \delta, \varepsilon, \zeta$ - веса, позволяющие контролировать важность каждого показателя, $RuSimScore \in [0, 1]$.

В результате применения этой функции авторам решения удалось улучшенить оценку SARI на тестовых данных на 0.6 в сравнении с результатами, полученными той же моделью, но без использования $RuSimScore$ (с 38.6 до 39.2).

В решении[31], занявшем третье место, как фильтрация тренировочных данных, так и выбор кандидатов выполнялись с помощью четырех показателей - косинусного сходства, ROUGE-L и длины исходного предложения и кандидата в токенах. Однако, в отличие от рассмотренного выше решения, вместо объединения четырех показателей в агрегат, здесь выбор лучшего кандидата осуществлялся путем обучения классификатора «Случайный лес», где перечисленные показатели использовались в качестве признаков. Данных для сравнения с результатами той же модели, но без использования этих признаков, не предоставлено.

Выводы из классификации решений

Таким образом, решения, использующие глубокое обучение, являются наиболее подходящими для поставленной задачи. При этом на качество модели сильное влияние оказывает использование дополнительных показателей для фильтрации обучающих данных или для выбора наиболее подходящего упрощения из созданных.

Рассмотренные метрики оценки качества упрощения, такие как SARI и SAMSA, показывают высокую корреляцию с оценкой человеком и позволяют применять технику обучения без учителя. Чтобы определенить оптимальные дополнительные параметры для моделей, использующих глубокое обучение, необходимо провести дополнительное сравнение решений по этим метрикам.

Заключение

В результате выполнения работы были проведены анализ предметной области и обзор существующих методов решения задачи упрощения текстов. Также были рассмотрены автоматические способы оценки качества упрощения, сформулированы критерии выбора подходящего решения и проведена классификация методов.

На основе классификации был сделан выбор в пользу решений, использующих глубокое обучение. Эти методы, во-первых, решают именно задачу упрощения текстов, а не задачи суммаризации или сокращения, во-вторых, наиболее комплексно упрощают текст, учитывая его лексическую и структурную составляющие.

Был сделан вывод, что выбор из методов глубокого обучения оказывает меньшее влияние на конечный результат, чем использование дополнительных показателей для фильтрации обучающих данных или для выбора наиболее подходящего упрощения из созданных, и что выбор оптимальных параметров требует дополнительного исследования.

Литература

1. Liu Jun, Matsumoto Yuji. Simplification of Example Sentences for Learners of Japanese Functional Expressions // Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016). 2016. 12. С. 1–5.

2. Evans Richard, Orăsan Constantin, Dornescu Iustin. An evaluation of syntactic simplification rules for people with autism // Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR). Association for Computational Linguistics, 2014. 04. С. 131–140.

3. Finegan-Dollak Catherine, Radev Dragomir. Sentence simplification, compression, and disaggregation for summarization of sophisticated documents // Journal of the Association for Information Science and Technology. 2015. 10. Т. 67. С. 2437–2453.

4. Shardlow Matthew. A Survey of Automated Text Simplification // International Journal of Advanced Computer Science and Applications. 2014. 01. Т. 4.

5. Siddharthan Advaith. A survey of research on text simplification // ITL – International Journal of Applied Linguistics. 2014. Т. 165. С. 259–298.

6. MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases / L. Martin, A. Fan, Éric de la Clergerie et al. // arXiv preprint arXiv:2005.00352. 2021.

7. Fisher Douglas, Frey Nancy, Lapp Diane. Text Complexity: Raising Rigor in Reading. Newark, DE: International Reading Association., 2012. с. 212.

8. Лингвистическая сложность учебных текстов / Александра Яковлевна Вахрушева, Марина Ивановна Солнышкина, Роман Владимирович Куприянов [и др.] // Вопросы журналистики, педагогики, языкознания. 2021. апр. Т. 40, № 1. С. 89–99.

9. Zhu Zhemin, Bernhard Delphine, Gurevych Iryna. A Monolingual Tree-based Translation Model for Sentence Simplification // Coling 2010 - 23rd

International Conference on Computational Linguistics, Proceedings of the Conference. 2010. 08. Т. 2. С. 1353–1361.

10. Sikka Punardeep, Mago Vijay. A Survey on Text Simplification // arXiv preprint arXiv:2008.08612. 2020. 08. С. 1–15.

11. RuSimpleSentEval-2021 Shared Task: Evaluating Sentence Simplification for Russian / Andrey Sakhovskiy, Alexandra Izhevskaya, Alena Pestova [и др.] // Proceedings of DIALOGUE 2021 conference. 2021. 06. С. 607–617.

12. М.С. Косолапов. Шкала Лайкерта (Ликерта). Социологический словарь / под ред. Г. В. Осипов. М.: Академический учебно-научный центр РАН-МГУ им. М.В. Ломоносова, НОРМА, НИЦ ИНФРА М, 2015. С. 372–373.

13. Paasche-Orlow Michael, Taylor Holly, Brancati Frederick. Readability Standards for Informed-Consent Forms as Compared with Actual Readability // The New England journal of medicine. 2003. 02. Т. 348. С. 721–6.

14. Optimizing Statistical Machine Translation for Text Simplification / Wei Xu, Courtney Napoles, Ellie Pavlick [и др.] // Transactions of the Association for Computational Linguistics. 2016. 12. Т. 4. С. 401–415.

15. Sulem Elior, Abend Omri, Rappoport Ari. Semantic Structural Evaluation for Text Simplification // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, 2018. С. 685–696.

16. See Abigail, Liu Peter J., Manning Christopher D. Get To The Point: Summarization with Pointer-Generator Networks // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics. С. 1073–1083.

17. Paetzold Gustavo, Specia Lucia. A Survey on Lexical Simplification // Journal of Artificial Intelligence Research. 2017. 11. Т. 60. С. 549–593.

18. NLP From Scratch: Translation with a Sequence to Sequence Network and Attention — PyTorch Tutorials 1.10.1+cu102 documentation. URL:

https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html (Дата обращения: 01-01-2022).

19. Валгина Н.С. Розенталь Д.Э. Фомина М.И. Современный русский язык: Учебник. 6-е изд., перераб. и доп. - М.: Логос, 2002. с. 528.

20. Chandrasekar Raman, Bangalore Srinivas. Automatic Induction of Rules for Text Simplification // Knowl.-Based Syst. 1997. 10. Т. 10. С. 183–190.

21. Siddharthan Advaith. Syntactic Simplification and Text Cohesion // Research on Language & Computation. 2004. 07. Т. 4.

22. Jonnalagadda Siddhartha R., Gonzalez Graciela. BioSimplify: an open source sentence simplification engine to improve recall in automatic biomedical information extraction // AMIA Annual Symposium proceedings. AMIA Symposium. 2010. Т. 2010. С. 351–5.

23. Barlacchi Gianni, Tonelli Sara. ERNESTA: A Sentence Simplification Tool for Children's Stories in Italian // Computational Linguistics and Intelligent Text Processing / под ред. Alexander Gelbukh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. С. 476–487.

24. Alva-Manchego Fernando, Scarton Carolina, Specia Lucia. Data-Driven Sentence Simplification: Survey and Benchmark // Computational Linguistics. 2020. Т. 46. С. 1–87.

25. Aluisio Sandra, Gasperin Caroline. PorSimples: Simplification of Portuguese Texts Fostering Digital Inclusion and Accessibility // Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas. 2010. 06.

26. A Review on Deep Learning Techniques Applied to Semantic Segmentation / Alberto Garcia-Garcia, Sergio Orts, Sergiu Oprea [и др.] // arXiv preprint arXiv:1704.06857. 2017. Т. abs/1704.06857.

27. Exploring Neural Text Simplification Models / Sergiu Nisioi, Sanja Stajner, Simone Ponzetto [и др.] // ACL. 2017. 01. С. 85–91.

28. Zhang Xingxing, Lapata Mirella. Sentence Simplification with Deep Reinforcement Learning // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017. С. 584–594.

29. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee [и др.] // arXiv preprint arXiv:1810.04805. 2019. 05.

30. Orzhenovskii Mikhail. RuSimScore: unsupervised scoring function for Russian sentence simplification quality. 2021. 06. С. 524–532.

31. Shatilov A. A., Rey A. I. Sentence simplification with ruGPT3. 2021. 06. С. 618–625.