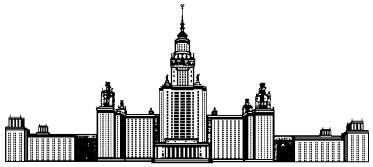
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«Ранжирование текстовых документов на основе оценок когнитивной сложности текста»

Выполнил:

студент 4 курса 417 группы

Еремеев Максим Алексеевич

Научный руководитель:

д.ф-м.н., профессор РАН

Воронцов Константин Вячеславович

Москва, 2020

Содержание

Аннотация

В данной работе описан подход к оцениванию когнитивной сложности текста на разных уровнях языка, и их использование для решения задачи ранжирования. В отличие от индексов удобочитаемости, которые основаны на комбинации текстовых статистик, данная работа предлагает обобщенный психофизиологический подход, позволяющий оценивать сложность на фонетическом, морфологическом, лексическом, синтаксическом и дискурсивном уровнях языка. Этого позволяет добиться использование референтного корпуса текстов и квантильного подхода для определения токенов с аномальной частотой. Собрав выборку размеченных пар документов русскоязычной Википедии, мы также обучаем и исследуем линейную комбинацию моделей со всех уровней языка. Полученные модели применяются для в модификации алгоритма построения деревьев чтения — одного их методов ранжирования результатов разведочного поиска. Также, работа представляет интерактивный веб-ресурс TextComplexity.net, демонстрирующий работу предложенных алгоритмов оценивания сложности. Для проведения дальнейших исследований была разработана библиотека с открытым кодом Cognitive Complexity. Приведенные в работе результаты экспериментов показывают конкурентоспособность предложенных подходов.

1 Введение

Инструменты автоматического измерения сложности текстов были изначально предложены учителям для облегчения процесса выбора учебников, соответствующих уровню понимания учеников, и издателям, для оценки уровня удобочитаемости их статей. Множество *индексов удобочитаемости* было разработано для реализации этих задач. В своем большинстве, они основываются на агрегации синтаксических и лексических признаков текста. Наиболее известными примерами индексов удобочитаемости являются *Автоматический Индекс Удобочитаемости* (Automated Readability Index) [17], *Индекс Удобочитаемости Флеша* (Flesch-Kincaid readability test) [7], *Gunning Fog Index* [8], *SMOG формула* [12]. Все индексы удобочитаемости используют статистики, такие как суммарное число слов, среднее количество слов в предложении или количество слогов для количественного оценивания сложности текста. Комбинируя эти статистики, индексы удобочитаемости присваивают каждому документу *оценку сложности*. Например, Автоматический Индекс Удобочитаемости (ARI) имеет вид для документа d:

$$ARI(d) = 4.71 \times \frac{c}{w} + 0.5 \times \frac{w}{s} - 21.43, \tag{1}$$

где c — количество букв в документе d, w — количество слов, s — количество предложений в d.

Индексы удобочитаемости интерпретируемы и просты в имплементации. Однако, многие индексы удобочитаемости привязаны к системе школьных оценкок США, и их требуется адаптировать к каждому языку отдельно. Это существенно ограничивает количество возможных приложений.

Примером адаптированной оценки сложности для русского языка может служить индекс *Flesch Readability Ease* (FRE), представленный в работах И. Оборневой [14], где были найдены оптимальные константы в Индексе Удобочитаемости Флеша для русского языка. Полученная оценка имеет вид:

$$FRE(d) = 206.836 - (1.52 \times ASL) - (65.14 - ASW), \tag{2}$$

где ASL — среднее количество слов в предложении, ASW — среднее количество слогов в слове.

В 2018 в работе В. Соловьева [18] представлена новая формула удобочитаемости, созданная исключительно для русского языка.

Несколько попыток агрегации оценок сложности были предприняты в работах [15] и [4]. Первая конструирует метрику Coh-Metrix-PORT, включающую более пятидесяти различных индексов удобочитаемости для португальского языка. Вторая работа представляет метод оценивания ментального представления сложности текстов.

Оценки сложности текста имеют множество приложений. Так, например, [5] описывает методы анализа юридических документов на русском языке на основе индексов удобочитаемости.

Альтернативный подход к оцениванию сложности был предложен в 2007 году. В работе [2] обсуждаются психофизиологические (когнитивные) методы оценивания сложности текста. В частности, выделяются следующие предположения:

1. Любой текст представляет собой последовательность токенов (кодов) — частей конечного алфавита — букв, слогов, предложений, слов, и.т.д.

2. Во время чтения текста, нервная система человека декодирует токены последовательно на следующих уровнях языка:

   (a) Фонетическом, декодирование звуков устной и букв письменной речи

   (b) Морфологическом, декодирование сочетаний букв и частей слов

   (c) Лексическом, декодирование отдельных слов

   (d) Синтаксическом, декодирование связей между словами, предложений

   (e) Дискурсивном, понимание идей и фактов

   На каждом этапе нервная система декодирует последовательность токенов, преобразуя ее в последовательность следующего языкового уровня. Получаемые таким образом «алфавиты» усложняются с каждым этапом, а токены в них соответствуют более крупным единицам речи.

3. Декодирование происходит в различных зонах нервной системы, начиная со зрительного или слухового аппарата, заканчивая корой головного мозга. Каж-

дая зона отвечает за декодирование определенного токена. Завершив декодирование, зона переходит в состояние *рефрактерности*, и ей требуется время на восстановление. В процессе восстановления, зона не способна производить декодирование, и если соответствующий ей токен вновь встретится в тексте, для декодирования будет задействована другая зона. Такое перераспределение ресурсов нервной системы уменьшает эффективность нервной системы в целом на последующих этапах декодирования, что приводит к затруднению восприятия текста человеком, к повышению утомляемости.

4. В процессе эволюции язык приспосабливается к психофизиологическим и культурологическим особенностям популяции. Результатом естественного отбора и приспособления является неравномерность распределения токенов как по частоте встречаемости, так и по сложности их обработки. Чем чаще встречается данный код в речи, тем быстрее происходит его обработка. Например, на фонетическом этапе высокочастотные буквы «о», «а», «е», «н», «т» обрабатываются быстрее, чем низкочастотные «ш», «ф», «ц», «э», «ю». Приоритетное выделение быстрых вычислительных ресурсов мозга для обработки более частых токенов характерно для всех типов токенов на всех этапах обработки сигнала. Пример: рис. 1.

5. Если частота токена превышает комфортную (обусловленную эволюцией нервной системы), то человек испытывает нагрузку при его расшифровке.

6. Если сложность одного токена превышает комфортную, то человек испытывает нагрузку при его декодировании (пример — слишком длинное слово).

7. Количество аномально частых и сложных токенов являются характеристикой сложности текста.

Рис. 1: Пример аномальной частоты буквы «р». Здесь она встречается примерно в четыре раза чаще, чем в среднем в русском языке.

В [2] авторы предлагают считать комфортное значение частоты (сложности) токена как квантиль эмпирического распределения частот (сложностей) этого токена, построенного по большой коллекции несложных текстов. Такая коллекция называется *референтным корпусом*. Авторы рассматривают фонетический уровень языка, ограничившись алфавитом букв. Работа [3] предлагает лексическую модель, предполагая, что сложность отдельного слова задается его длиной. Модель дискурсивного уровня, представленная в [24], считает сложностью предложения количество специальных слов-связок в предложении.

Данная работа, базируясь на предположениях выше, предлагает модели сложности текстов для морфологического, лексического и синтаксического уровнях языка, а также объединенную модель сложности текста на всех уровнях. Приложением данной теории является ранжирование результатов *разведочного поиска* в образовательных или в целях редактирования текстов [11, 22, 20]. В разведочном поиске пользователю необходимо указать, с каких документов нужно начать изучение новой темы, постепенно продвигаясь к более узкоспециализированным текстам. Одной из разновидностей данной задачи является задача построения *порядка чтения* (Reading Order). Это альтернативный подход к потреблению текстовой информации, который вместо использования обычных ранжированных списков располагает релевантные документы в виде деревьев знаний [10].

В разделе 2 данной работы обсуждается обобщенная модель когнитивной сложности текста, формализуя понятия «сложность» и «частотность» в разделе 3. Модели для конкретных видов токенов представлены в разделе 4, и, переходя к агрегированной модели в разделе 6, работа предварительно описывает собранный путем краудсорсинга набор данных в разделе 5. Описание экспериментов содержится в секции 7, в то время как секции 8 и 9 описывают приложения предложенной теории: модификацию стандартного алгоритма *Reading Order* и веб-ресурс *TextComplexity.net* вместе с библиотекой с открытым кодом *Cognitive Complexity*. Раздел 10 подводит итог, собирая все результаты. В приложении А можно найти примеры эмпирических распределений, восстановленных по разным референтным корпусам для разных моделей сложности.

2 Обобщенная модель сложности текста

Обобщённая модель нагрузки восприятия основана на предположении, что каждый токен в тексте имеет свою сложность обработки. Если она оказывается аномально высокой по сравнению с «привычной» сложностью обработки токенов того же типа, то мозг испытывает избыточную нагрузку. Суммирование этих нагрузок по всему тексту позволяет оценить его когнитивную сложность. «Привычная» сложность обработки токенов обусловлена эволюционным развитием языка в культурно-историческом контексте. Для её оценивания строится эмпирическое распределение значений сложности токенов по референтному корпусу текстов.

Формальные определения. Пусть d — произвольный документ, состоящий из токенов $x_1, \ldots, x_n$ из фиксированного конечного алфавита A_h. Здесь, h означает уровень языка, т.е. фонетический, морфологический, лексический, синтаксический или дискурсивный. Токенами, входящими в алфавит, могут быть буквы, слова, предложения, и.т.д. Обозначим через c_i *когнитивную оценку сложности*, а за w_i — величину нагрузки для токена x_i. Тогда назовем *оценкой сложности документа* сумму нагрузок токенов, имеющих аномальную сложность.

Референтный корпус текстов. Оценок сложности считаются относительно *референтного корпуса* — множество документов средней сложности. Пусть задан референтный корпус текстов из n_0 документов с длинами $n_1, \ldots n_{n_0}$. Тогда построим для каждого токена a эмпирическое распределение значений сложности c_i по всем позициям i, на которых находится данный код, $x_i = a$, рис. 2. Для каждого токена a определим γ-квантиль эмпирического распределения сложности. Это такое число $C_\gamma(a)$, что доля кодов $x_i = a$ в референтном корпусе, для которых $c_i \leq C_\gamma(a)$, равна γ:

$$\sum_{d=1}^{n_0} \sum_{i=1}^{n_d} [x_i = a]\,[c_i \leq C_\gamma(a)] = \gamma \sum_{d=1}^{n_0} \sum_{i=1}^{n_d} [x_i = a].$$

Параметр γ задается таким образом, чтобы сложность c_i выше порогового значения $C_\gamma(a)$ можно было считать аномально высокой. Например, $\gamma = 0.9$ означает, что лишь 10% токенов в референтном корпусе имеют сложность выше пороговой.

В экспериментах использованы два типа референтных корпусов — русскоязычная Википедия (1.5 миллионов текстов) и коллекция открытых текстов «Ноосфера»

(noosphere.ru) (200 тысяч текстов). Первая, являясь энциклопедией, состоит из специфичных, более научных статей, в то время как последняя включает разнообразные типы текстов, в том числе произведения художественной литературы.

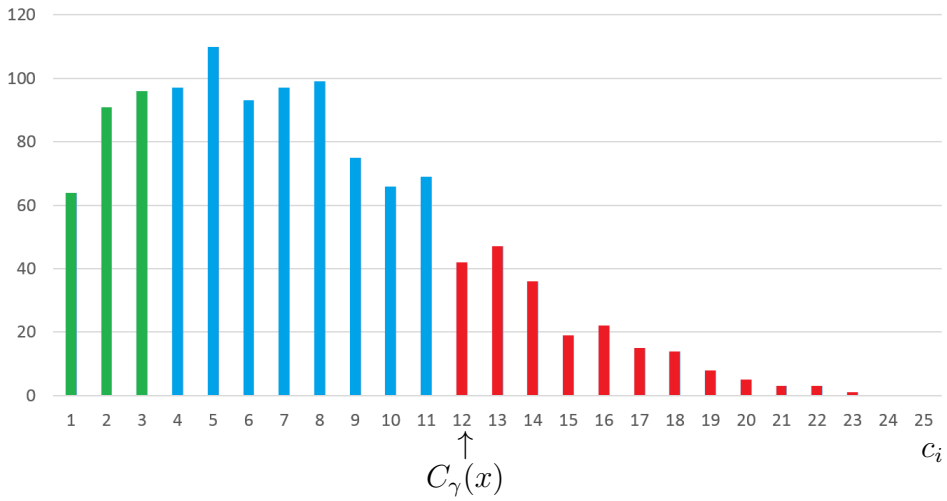


Рис. 2: Пример эмпирического распределения сложности кода. Высота i-й колонки гистограммы показывает, сколько раз код сложности i встретился в тексте. Красная зона соответствует аномально высокой сложности, вызывающей дополнительную нагрузку восприятия. Зеленая зона соответствует низкой сложности. Синяя зона согласуется с эволюционной моделью.

Оценка сложности документа. Таким образом, сложность $W(d)$ документа d получается суммированием сложностей c_i всех аномально сложных токенов из d с весами, равными нагрузкам w_i.

$$S(d) = \sum_{i=1}^{n} w_i \left[c_i > C_\gamma(x_i)\right], \tag{3}$$

где $[\,]$ — нотация Айверсона (т.е. [истина] $= 1$, [ложь] $= 0$), n — число токенов из A_h в документе d.

Значение $S(d)$ зависит, вообще говоря, от длины документа и может быть сколь угодно большим. Поэтому вводится нормированная когнитивная сложность документа d:

$$W(d) = \frac{\sum\limits_{i=1}^{n} w_i \left[c_i > C_\gamma(x_i)\right]}{\sum\limits_{i=1}^{n} w_i}, \tag{4}$$

Если $w_i = 1$, то $S(d)$ равно числу токенов с избыточной нагрузкой, $W(d)$ равно доле таких токенов. Если текст имеет примерно ту же сложность, что и референтный корпус, то $W(d) \approx 1 - \gamma$.

В обобщенной модели не уточняется, как именно определяется сложность c_i и нагрузка w_i. В общем случае нагрузка w_i должна быть неотрицательной величиной, монотонно неубывающей с ростом сложности c_i. Сложность c_i также определена с точностью до произвольной монотонно возрастающей функции μ (возможно, своей для каждого токена), поскольку условия $c_i > C_\gamma(x_i)$ и $\mu(c_i) > \mu(C_\gamma(x_i))$ эквивалентны, а оценки c_i используются только для вычисления квантили эмпирического распределения сложности и сравнения значений сложности с этой квантилью.

Если токен x_i не входил в референтный корпус, то $C_\gamma(x_i)$ полагается равным $-\infty$, тем самым модель всегда считает такой токен аномально сложным.

Задавая алфавит A_h и способ вычисления параметров c_i и w_i, можно строить различные модели нагрузки восприятия текста, затем выбирать лучшую модель по заданному количественному критерию.

3 Функции сложности отдельных токенов

В данном разделе вводятся два основных подхода измерения сложности отдельного токена. Они основываются на предположениях 6 и 7.

3.1 Частотная функция

Основной метод оценки когнитивной сложности одного токена, рассматриваемый в данной работе — частотный. Он основан на предположениях психофизиологической теории 3-5.

Расстояние до предыдущего вхождения токена. Пусть r_i — расстояние до предыдущего вхождения токена x_i до его текущего вхождения в текст:

$$\ldots \quad \boxed{x_{i-r_i} = a} \quad \underbrace{x_{i-r_i+1} \quad x_{i-r_i+2} \quad \ldots \quad x_{i-2} \quad x_{i-1} \quad \boxed{x_i = a}}_{r_i} \quad \ldots$$

Формально,

$$r_i = \min_{1 \le j < i} \{i - j \mid x_i = x_j\}. \tag{5}$$

Если токен x_i вошел в документ d впервые на позиции i, то у него нет предыдущего вхождения, и, следовательно, r_i не определено. В таком случае, r_i доопределяется так, чтобы сумма всех r_i вхождений одного и того же токена $x_i = a$ была равна n — длине документа. Можно представить, что перед началом текста стоит его дубликат.

Для примера, пусть A_h состоит из букв. Изначально, возможно определить значения r_i только для семи позиций. После доопределения всем токенам гарантированно присваиваются значения r_i:

токен	г	е	р	о	й	н	а	ш	е	г	о	в	р	е	м	е	н	и
r_i исходное	–	–	–	–	–	–	–	–	7	9	7	–	10	5	–	2	11	–
r_i доопред.	9	4	8	11	18	7	18	18	7	9	7	18	10	5	18	2	11	18

Рис. 3: Пример: r_i и доопределенные r_i для фразы «герой нашего времени».

Период блокировки. С точки зрения нейрофизиологии обработчиком токена является зона нервной системы, состоящая из большого числа нейронов. После обработки токена они ещё некоторое время остаются в состоянии рефрактерности. В этот период зона заблокирована и не способна обработать следующее появление данного токена.

Рассмотрим несколько способов оценить *период блокировки* $R(a)$.

Для начала определим $R(a)$ как среднее расстояние r_i по всем вхождениям токена $x_i = a$ в референтный корпус, состоящий из n_0 документов с длинами $n_1, \dots n_{n_0}$. Оно равно инвертированной частоте токена a:

$$R(a) = M(a) = \frac{\sum_{d=1}^{n_0} \sum_{i=1}^{n_d} r_i[x_i = a]}{\sum_{d=1}^{n_0} \sum_{i=1}^{n_d} [x_i = a]} = \frac{\sum_{d=1}^{n_0} n_d}{\sum_{d=1}^{n_0} \sum_{i=1}^{n_d} [x_i = a]}.$$

Данная оценка может оказаться сильно завышенной. Нейрон имеет период рефрактерности порядка 100–300 мс. Обычная скорость чтения 700 знаков в минуту даёт оценку скорости обработки одной буквы около 80 мс. Это означает, что период блокировки едва ли превышает время обработки 4-х буквенных кодов. Однако

мы не знаем, сколько нейронов образуют зону обработки одного токена, насколько дольше зона восстанавливается по сравнению с отдельным нейроном, и насколько сложно могут быть устроены обработчики следующих этапов декодирования. Кроме того, известно, например, что слова распознаются корой головного мозга скорее целиком, чем по отдельным буквам.

Перечисленные трудности приводят к параметрическим моделям, в которых период блокировки некоторым разумным образом сокращается.

Вместо $M(a)$ можно взять β-квантиль $M_\beta(a)$ эмпирического распределения расстояний $\{r_i\colon x_i = a\}$ в референтном корпусе. Значение параметра $\beta = 0.5$ соответствует медиане этого распределения. Предположение, что период блокировки адаптирован к повышенным нагрузкам, соответствует значениям β от $1-\gamma$ до 0.5.

Для редко встречающихся токенов величина $M_\beta(a)$ может оказаться всё ещё слишком большой. Длительность периода блокировки можно ограничить сверху с помощью кусочно-линейного преобразования с параметром R_0

$$R(a) = \min\bigl\{M_\beta(a), R_0\bigr\},$$

либо с помощью его гладкой аппроксимации снизу гиперболическим тангенсом:

$$R(a) = R_0 \tanh\bigl(M_\beta(a)/R_0\bigr). \tag{6}$$

Частотная функция сложности. Чтобы получить частотную модель нагрузки как частный случай обобщённой модели, параметры c_i и w_i определяются через расстояния r_i.

Сложность c_i тем выше, чем меньше расстояние r_i по сравнению с *периодом блокировки* данного токена $R(x_i)$:

$$c_i = R(x_i) - r_i. \tag{7}$$

Нагрузку w_i определим как число обработчиков кода a, заблокированных к моменту обработки токена x_i, как показано на рис.1. Оно совпадает с числом вхождений токена $x_i = a$ в текстовый фрагмент $x_{i-R(a)+1}, \ldots x_i$, длительность которого равна периоду блокировки:

$$w_i = \sum_{j\,=\,i-R(x_i)+1}^{i} [x_j = x_i]. \tag{8}$$

Параметры модели β и R_0, как и общий для всех моделей параметр γ предполагается оптимизировать по внешнему критерию качества. В частности, для этого можно использовать данные лингвистических или нейрофизиологических экспериментов по сравнению сложности текстов, читаемых разными людьми. Подобные эксперименты рассмотрены в разделе 7.

3.2 Сложностная функция

В качестве альтернативы частотному подходу, вводятся сложностные функции нагрузки. Согласно предположению 6, нагрузка при декодировании токена может быть вызвана не только аномально высокой частотой некоторых токенов, но и аномальной сложностью их обработки. Токены могут различаться по сложностям их внутренней структуры или по сложности их взаимосвязи с контекстом. Излишняя нагрузка возникает, когда сложность токена превышает его привычную (эволюционно обусловленную) сложность.

В сложностном подходе, как в частном случае обобщенного подхода, предполагается, что алфавит A_h состоит из единственного токена $A_h = \{a\}$. То есть, будем различать не сами токены, а только их сложности. Сложности токенов определяются их лингвистическими свойствами, и каждый токен имеет ровно одно возможное значение сложности.

Таким образом, будет построено одно эмпирическое распределение значений сложности для всех токенов из референтного корпуса текстов. В таком случае, $C_\gamma(x_i) = C_\gamma$, а модель 3 будет иметь следующий вид:

$$S(d) = \sum_{i=1}^{n} w_i \left[c_i > C_\gamma\right] \tag{9}$$

Значения нагрузок w_i определять по-разному:

$w_t = 1$;

$w_i = R(x_i)$;

$w_i = M(x_i)$.

В экспериментах же был использован простейший вариант — $w_i = 1$.

4 Рассматриваемые модели

Предложенная обобщённая модель позволяет порождать новые модели нагрузки восприятия, задавая алфавит токенов и способ вычисления параметров c_i для каждой позиции i. В то же время, современные средства компьютерной лингвистики предоставляют богатые возможности для формирования алфавитов токенов и характе.тик их сложности. Для этого могут использоваться готовые средства морфологического, лексического, синтаксического и семантического анализа. В данном разделе предлагаются модели для различных уровней языка.

4.1 Фонетический уровень

На фонетическом уровне токенами являются отдельные буквы (рис. 3). *Частотная модель букв* рассматривалась в [2] в предположении, что буквы, имеющие аномальную частоту вызывают дополнительную нагрузку нервной системы. Пример построенного эмпирического распределения для корпуса Википедии изображен на рис. 11.

4.2 Морфологический уровень

Эксперименты с перестановками букв в словах показывают, что наш мозг распознает слова не по буквам, а по коротким последовательностям из n букв, n-граммам (рис. 4). Аномальное употребление отдельных n-грамм должно приводить к нагрузкам восприятия текста.

На морфологическом уровне в качестве токенов используются n-граммы, формализуя предположение выше. Обычно n-граммы выделяют из текста «внахлест», то есть каждая буква слова (за исключением и $n-1$ последних) является началом новой n-граммы, рис. 5. Возможны два варианта формирования токенов: либо брать исходные n-граммы, либо переставлять в них буквы в алфавитном порядке, чтобы порядок букв не учитывался. В первом случае биграммы «ер» и «ре» различаются, во втором случае они становятся эквивалентными. Второй вариант позволяет сократить алфавит токенов, избежав его избыточной детализации и вынужденного увеличения объёма референтного корпуса. Если в исходном алфавите 30 букв, то

По рзелульаттам илссеовадний одонго анлигйского унвиертсиета, не иеемт занчения, в кокам пряокде рсапожолены бкувы в солве. Галвоне, чотбы преавя и псолендяя бквуы блыи на мсете. Осатьлыне бкувы мгоут селдовтаь в плоонм бсепордяке, все рвано ткест чтаитсея без порбелм. Пичрионй эгото ялвятеся то, что мы не чиатем кдаужю бкуву по отдльенотси, а все солво цликеом.
По результатам исследований одного английского университета, не имеет значения, в каком порядке расположены буквы в слове. Главное, чтобы первая и последняя буквы были на месте. Остальные буквы могут следовать в полном беспорядке, все равно текст читается без проблем. Причиной этого является то, что мы не читаем каждую букву по отдельности, а все слово целиком.

Рис. 4: Пример, показывающий, что распознавание слов происходит скорее по n-граммам, чем по отдельным буквам, причем порядок букв в n-граммах большого значения не имеет.

алфавит биграмм ($n = 2$) состоит из 900 элементов в первом случае, и сокращается до 435 элементов во втором. Алфавит триграмм ($n = 3$) состоит соответственно из 27000 или 4060 элементов. Нужен очень большой референтный корпус текстов, чтобы по редким триграммам накопился достаточный объём статистики.

1-граммы	г	е	р	о	й	н	а	ш	е	г	о	в	р	е	м	е	н	и

n-граммы, сохраняющие порядок букв

2-граммы	ге	ер	ро	ой		на	аш	ше	ег	го		вр	ре	ем	ме	ен	ни	
3-граммы	гер	ero	рой			наш	аше	шег	его			вре	рем	еме	мен	ени		

n-граммы, не сохраняющие порядок букв

2-граммы	ге	ер	ор	йо		ан	аш	еш	ге	го		вр	ер	ем	ем	ен	ин	
3-граммы	гер	еор	йор			анш	аеш	геш	гео			вер	емр	еем	емн	еин		

Рис. 5: Представление текста «герой нашего времени» в виде последовательности униграмм (букв), биграмм и триграмм, с сохранением и без сохранения порядка букв в n-граммах.

Альтернативные методики формирования токенов, занимающих промежуточное положение между буквами и словами, заключается в разбиении слов либо на слоги. Такой подход является сбалансированным, так как слоги не фиксированы по длине, но являются самостоятельными произносительными единицами. В экспериментах рассматриваются две модели: *частотная морфологическая модель с токенами-слогами с учетом и без учета порядка букв* (в последнем случае, буквы в слогах сортируются в алфавитном порядке).

Соответствующие эмпирические распределения по корпусу Википедии приведены на рис. 12 и 13.

4.3 Лексический уровень

На лексическом уровне токенами являются отдельные слова или словосочетания. На данном уровне словоформы считаются эквивалентными, поэтому в качестве предобработки проводится лемматизация слов (приведение слов к нормальной форме).

Частотная модель. Главное предположение *частотной лексической модели* состоит в том, что чем чаще встречается одно и то же слово в тексте, тем большую нагрузку оно оказывает на нервную систему человека. Для каждого токена формируется эмпирическое распределение частот по референтному корпусу и определяется уровень аномальной частоты.

Примеры таких распределений для токена «МАТЕАМАТИКА» для референтных корпусов Википедии и «Ноосфера» приведены на рис. 14 и 15.

Сложностная модель длины слова. Сложностная модель на лексическом уровне была описана в [3]. В качестве оценки сложности слова выбиралась его длина. Таким образом, слово считалось аномально сложным, если оно аномально длинное. Строя эмпирическое распределение всех длин слов в референтном корпусе, получаем *сложностную модель длины слова*.

Сложностная модель терминов. *Сложностная модель терминов* построена в предположении, что чем реже слово встречается в референтном корпусе, тем более специфичным оно является, а узкоспециализированные термины оказывают нагрузку при их декодировании.

Функция сложности является убывающей функций значения $count(x_i)$, которое равно количеству вхождений слова x_i в референтный корпус. В экспериментах использована следующая функция:

$$c_i = -count(x_i) \tag{10}$$

В такой постановке строится единой эмпирическое распределение сложностей по всему референтному корпусу, а порог аномальной сложности устанавливается как γ-квантиль найденного распределения.

4.4 Синтаксический уровень

Токены синтаксического уровня — предложения. Для деления текста на предложения и построения деревьев синтаксического анализа была использована библиотека UDPipe [19]. С помощью дерева синтаксического анализа можно извлекать зависимости между словами (например, в словосочетании «красный шарик», слово «красный» является зависимым, а «шарик» — основным), главные и второстепенные члены предложения, типы членов предложения (подлежащее, сказуемое, дополнение, и.т.д.) и частей речи (глагол, существительное, прилагательное, и.т.д.). Используя полученную информацию, предлагается две модели сложности.

Сложностная модель. Разумно предположить, что аномально длинные синтаксические связи могут приводить к нагрузке на нервную систему при декодировании. Как это делалось для аномально длинных слов, *сложностная синтаксическая модель* определяет функцию сложности как максимальную длину синтаксической связи в предложении.

Частотная модель. Другой подход заключается в рассмотрении предложения как специального набора токенов — синтгам. Каждая синтгама соответствует одному слову, однако само слово отбрасывается, оставляя лишь информацию о его части речи и члене предложения. Полученные структуры имеют характерные неравномерные распределения по частотам в текстах языка. Поэтому, разумно предположить, что аномально частое употребление отдельных структур должно приводить к нагрузке восприятия.

Формально, пусть A_h — декартово произведения множеств PoS — множество всевозможных частей речи, и SP — множество всевозможных типов членов предложения. Тогда каждый токен $a \in A_h$ является парой (p, s), где $p \in PoS$ и $s \in SP$ — часть речи и тип члена предложения соответственно. Такие пары называются *синтгамами*.

Применяя частотную функцию сложности **??** к токенам такого типа, имеем (*синтаксическую модель синтгам*).

Примеры эмпирических распределений изображены на рис. 16 и 17.

4.5 Дискурсивный уровень

Предложенная в [24] *сложностная модель слов-связок*, является моделью дискурсивного уровня. На этом уровне оценивается осмысленность текста, его связность и последовательность.

Токенами на данном уровне являются предложения, а сложностью токена — количество распространенных слов-связок (например, «который», «из-за того что», «с тех пор как», и.т.д.) русского языка. Модель основана на предположении, что обработка предложений с аномально большим количеством логических связок вызывает дополнительную нагрузку. Всего подобных связок — 135.

5 Набор данных

Набор данных для валидации результатов и обучения агрегированной модели, был получен с помощью краудсорсинговой платформы Яндекс.Толока. Асессорам было предложено разметить 10 тысяч пар документов русскоязычной Википедии, выбрав какая из двух статей требует больше усилий для понимания. Интерфейс состоял из двух ссылок на соответствующие статьи и четырех вариантов ответа: «Левая статья сложнее», «Правая статья сложнее», «Обе статьи имеют одинаковую сложность», «Невозможно определить». Последний вариант следовало выбирать в случае, если статьи имеют разную тематику и их невозможно сравнить. Интерфейс задания приведен на рис. 6. Полная инструкция, предложенная асессорам, приведена на рис. 7.

Тематическое моделирование. Документы для разметки выбирались из области физики, математики, медицины и информатики. Для такого отбора документов был использован подход тематического моделирования [9]. А именно, подход к тематическому моделированию Additive Regularization of Topic Models (ARTM) [1, 21].

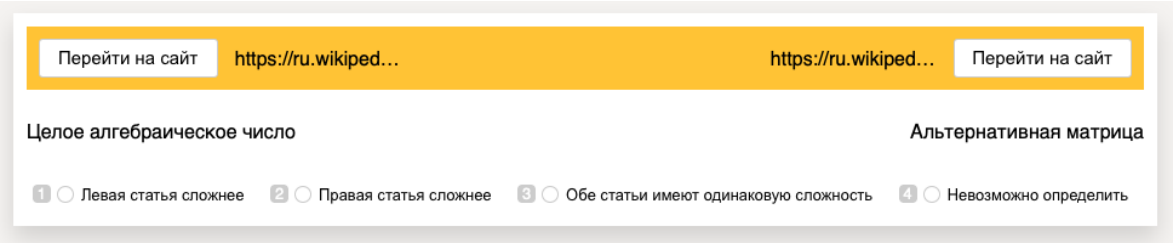


Рис. 6: Интерфейс задания по разметке для асессоров на Яндекс.Толока.

В данном задании Вам предлагается сравнить сложность двух статей Википедии. Вы увидите их заголовки и ссылки для перехода. Перейдите на обе страницы и полностью прочитайте их.

Какая из двух статей показалась вам более сложной – потребовала больше усилий для её понимания, содержала больше незнакомых терминов? Прочитав статью, запомните или запишите приблизительную оценку, какой процент содержимого этой статьи был трудным для понимания. Если статьи имеют одну тематику, но Вы затрудняетесь сразу определить, какая из них сложнее, то выбирайте вариант «Статьи имеют одинаковую сложность».

Мы старались подбирать пары статей так, чтобы они были из одной области знания. Если они всё же оказались из совершенно разных областей, и Вы лучше разбираетесь в одной из них, то выбирайте вариант «Невозможно определить». Цель нашего эксперимента – не сравнить Ваши познания, а выработать объективную оценку сложности текстов.

Рис. 7: Инструкция по разметке статей для асессоров на Яндекс.Толока.

Преимущество использования тематического моделирования для отбора статей для разметки перед использованием категорий из Википедии состоит в обеспечении тематической моделью мягкой кластеризации документов по темам. Так, каждому документу присваиваются не метки конкретных кластеров (тем), а стохастический вектор вероятностей принадлежности документа всем кластерам (тематические векторы). Это дает возможность не просто выбирать документы для разметки из одной темы, но и подбирать их так, чтобы они были тематически похожими друг на друга. Для проверки тематической схожести использовалась косинусная мера между тематическими векторами.

Пусть документ d_1 имеет тематический вектор θ_1, документ d_2 — θ_2. Тогда будем считать, что документы d_1, d_2 *тематически ε-близки*, если

$$cos(\theta_1, \theta_2) = \frac{(\theta_1, \theta_2)}{||\theta_1||||\theta_2||} > \varepsilon, 0 \leq \varepsilon < 1, \tag{11}$$

где (θ_1, θ_2) — скалярное произведение векторов θ_1, θ_2, $||\theta||$ — L_2-норма вектора θ.

Документы d_1 и d_2 могут быть выбраны для разметки, если они тематически ε-близки. В экспериментах была посторена тематическая модель русскоязычной Википедии на 70 тем. Метод ARTM поддерживает построение мультимодальных моделей [13]. Для модели на Википедии были выбраны модальность слов и биграмм (словосочетаний). Для сокращения словаря биграмм был использован метод выделения коллокаций TopMine [16]. При отборе пар документов по тематической близости использовалось значение $\varepsilon = 0.9$. Также, при отборе пар использовалось условие близости документов по длине. Пример отобранных и размеченных пар приведен в таблице 1.

Название левой статьи	**Название правой статьи**	**Какая статья сложнее?**
Матрица	Тензор	Правая
Рациональное число	Дробь (математика)	Левая
Протон	Нейтрон	Одинаковая сложность
Mac OS X	Выпуклая оболочка	Невозможно определить

Таблица 1: Примеры размеченных пар статей Википедии.

Корректность разметки. Каждая пара была размечена дважды, чтобы избежать ошибок, связанных с человеческим фактором. Предполагается, что пара была *корректно размечена*, если разметки двух асессоров не противоречивы, т.е. один из них утверждает, что левая статья сложнее, а второй, наоборот, что правая. Все остальные пары, а также пары с хотя бы одной меткой «Невозможно определить» выкидывались из финального набора данных. Такой отфильтрованный набор, в итоге, состоял из 8,5 тыс. размеченных пар документов.

Итоговая выборка здесь и далее обозначается как $D = \{(d, d') \mid d' \text{ сложнее чем } d)\}$. Для сокращения выкладок, обозначим принадлежность $(d, d') \in D$ как $d \prec d'$.

6 Агрегированная модель

Имея набор данных, описанный в разделе 5, и модели из раздела 4, возможно построить обучаемую модель сложности, агрегирующую все ранее представленные модели. Полученная модель будет объединять модели сложности со всех уровней языка.

Размер набора данных не позволяет обучать модель с большим числом параметров, поэтому в качестве базовой модели используется линейная регрессия.

$$W(d,\alpha) = \sum_{k=1}^{K} \alpha_k W_k(d), \quad \alpha_k \geq 0, \tag{12}$$

где вектор α является решением оптимизационной задачи:

$$\sum_{d \prec d'} \mathscr{L}\big(\underbrace{W(d',\alpha) - W(d,\alpha)}_{\text{pair-wise margin}}\big) \to \min_{\alpha}, \tag{13}$$

где $\mathscr{L}(M)$ гладкая невозрастающая функция отступа M.

Данный вид функционала выбран исходя из типа задачи. Действительно, решаемая задача является задачей типа learning to rank, так как на вход подаются ранжированные пары документов. В классе таких задач, функционал вида (12) является стандартным.

Для избежания переобучения используется тип регуляризации ElasticNet [23], комбинирующий L1 и L2 регуляризаторы:

$$\frac{1}{2|D|} \sum_{d \prec d'} \mathscr{L}\big(W(d',\alpha) - W(d,\alpha)\big) + \lambda \left((1-\zeta) \sum_{k=1}^{K} \alpha_k^2 + \zeta \sum_{k=1}^{K} |\alpha_k| \right) \to \min_{\alpha}, \tag{14}$$

где ζ — параметр, смешивающий Ridge ($\zeta = 0$) and Lasso ($\zeta = 1$) методы, λ контролирует вклад регуляризации в модель в целом. Для функции $\mathscr{L}$ были рассмотрены три варианта гладких невозрастающих функций:

Отрицательный квардрат ошибки (SE): $\mathscr{L}(M) = -M^2$,

Отрицательная сигмоида: $\mathscr{L}(M) = -\sigma(M)$, где $\sigma(x) = \frac{1}{1+\exp x}$ — сигмоидная функция,

Отрицательная абсолютная ошибка (AE): $\mathscr{L}(M) = -|M|$.

Результаты валидации всех моделей приведены в следующем разделе «Эксперименты».

7 Эксперименты

Эксперименты проводились для референтных корпусов русскоязычной Википедии и коллекции «Ноосфера». Качество измеряется на описанном в разделе 5 наборе данных из размеченных пар документов русскоязычной Википедии. В качестве метрики качества взята точность (accuracy), т.е. отношение правильно выбранных моделью более сложных статей к общему числу пар.

$$\text{accuracy(c)} = \frac{\sum\limits_{d \prec d'} [c(d) < c(d')]}{|D|} \tag{15}$$

Во всех экспериментах была использована модель 4. Для оценки качества агрегированной модели мы используем кросс-валидацию с четырьмя фолдами.

7.1 Отдельные модели

Описанные модели в разделе 4 сравниваются как с различными индексами удобочитаемости (стандартные бэйзлайны), так и между собой на разных уровнях языка. Результаты приведены в таблице 2. В качестве гиперпараметров использована $\gamma = 0.95, \beta = 0.6$.

Выводы.

1. Частотная лексическая модель демонстрирует наилучшее качество среди всех представленных моделей.

2. Все модели, основанные на квантильном подходе, превосходят индексы удобочитаемости, так как последние являются существенно более простыми и часто выдают похожую оценку удобочитаемости для пары статей Википедии.

3. Частотная морфологическая модель без учета порядка превосходит по качеству аналогичную, учитывающую порядок. Это подтверждает гипотезу об устойчивости получаемых распределений за счет сокращения размера словаря.

4. Лексические и морфологические модели показывают лучшее качество по сравнению с остальными квантильными моделями. Таким образом, нагрузка при декодировании слов и их частей влияет сильнее всего на нервную систему.

Класс моделей	**Модель**	**Accuracy**
Индексы удобочитаемости	Automated Readability Index	50.5%
	Flesch-Kincaid Grade	44.7%
	Gunning FOG	44.4%
	Flesch Reading Ease	50.7%
	Dale–Chall	37.0%
	Linsear Write	45.2%
	Coleman-Liau	**52.1%**
Фонетические	Частотная (буквы)	62.5%
Морфологические	Частотная (с учетом порядка)	70.9%
	Частотная (без учета порядка)	73.1%
Лексические	Сложностная (длина слова)	42.4%
	Частотная	**75.0%**
	Сложностная	71.2%
Синтаксические	Сложностная (длина синтаксической связи)	62.0%
	Частотная (синтгамы)	64.2%
Дискурсивные	Сложностная (слова-связки)	62.5%

Таблица 2: Сравнение значений accuracy для индексов удобочитаемости и моделей различных уровней языка.

5. На синтаксическом уровне модели показывают примерно равный результат, что выдвигает гипотезу о том, что большинство сложностных моделей могут быть приближены частотными при правильном выборе токенов.

7.2 Агрегированная модель

Агрегированные модели сравниваются между собой в зависимости от выбранной функции отступа и с наиболее эффективными моделями каждого типа. Результаты приведены в таблице 3. Гиперпараметры в экспериментах устанавливались равными $\zeta = 0.5$ и $\lambda = 10$ для всех агрегированных моделей сложности.

Модель	Функция отступа	**Accuracy**
Coleman-Liau	-	52.1%
Морфологическая (отсортированные слоги)	-	73.1%
Лексическая (частотная)	-	75.0%
Синтаксическая (синтгамы)	-	64.2%
Дискурсивная	-	62.5%
Агрегированная модель	Отрицательное SE	**88.1%**
Агрегированная модель	Отрицательная сигмоида	84.6%
Агрегированная модель	Отрицательное AE	85.1%

Таблица 3: Сравнение агрегированных моделей сложности с различными функциями отступа с моделями разного типа.

Выводы.

1. Все агрегированные модели показывают существенный прирост качества. Это говорит о том, что сложность документа действительно складывается из нагрузки декодирования на отдельных уровнях, но не равным образом.

2. Все функции отступа показывают примерно равное качество, однако отрицательный средний квадрат ошибки является наиболее эффективным.

7.3 Референтный корпус «Ноосфера»

Для оценивания вклада референтного корпуса в результат работы моделей. Заменяя референтный корпус Википедии на референтный корпус Ноосферы, ожидается получить изменение качества, так как объем корпуса меньше, а природа текстов отличается. Оценки качества все еще замеряются на наборе данных, описанном в секции 5, состоящем из восьми с половиной тысяч размеченных пар статей русскоязычной Википедии. Результаты приведены в таблице 4.

Класс моделей	**Модель**	**Accuracy**
Фонетические	Частотная (буквы)	60.3%
Морфологические	Частотная (с учетом порядка)	69.2%
	Частотная (без учета порядка)	69.4%
Лексические	Сложностная (длина слова)	39.8%
	Частотная	**72.1%**
	Сложностная	66.9%
Синтаксические	Сложностная (длина синтаксической связи)	63.1%
	Частотная (синтгамы)	66.4%
Дискурсивные	Сложностная (слова-связки)	60.2%
Агрегированные	Агрегированная с отрицательным SE	79.1%

Таблица 4: Сравнение значений accuracy для моделей различных уровней языка, обученных на корпусе «Ноосфера».

Выводы. Сравнивая результаты в таблицах 4, 2 и 3 можно сделать следующие выводы:

1. Значения метрики качества для всех моделей ниже, чем в первом эксперименте. Это можно объяснить тем, что коллекция «Ноосфера» меньше, а эмпирические распределения менее стабильны. Также коллекция состоит из как из студенческих работ, так и произведений художественной литературы, а значит там не встречаются многие термины из Википедии. Это может приводить к завышенным оценкам сложности.

2. Синтаксические модели являются исключением, их качество возрастает. Такой эффект проявляется, отчасти, из-за вариативности синтаксических структур в корпусе, а также отсутствия большого количества формул, ссылок, и.т.д., которые могли остаться после обработки текста.

3. Наибольшее падение качества демонстрирует агрегированная модель. Это наводит на мысль, что обучать ансамбль разумно на размеченных парах из того же набора данных, на котором обучались основные модели.

8 Построение списков чтения

В качестве приложения предложенной теории оценивания сложности, в данной работе рассматривается задача генерации порядка чтения (Reading Order). Данная техника является техникой ранжирования результатов выдачи разведочного поиска, предлагающая пользователю изучать документы из поисковой выдачи в логическом порядке. Для решения такой задачи применимы методы оценивания сложности. Действительно, для освоения новой темы разумно начинать с более простых документов и двигаться к более сложным. Работа [10] предлагает алгоритм построения деревьев чтения, который можно улучшить, внедрив оценки сложности.

8.1 Постановка задачи

Порядок чтения $R(V, E)$ на коллекции документов D — это направленный ациклический граф, вершины которого соответствуют подмножеству документов D. Пусть вершине $v_i \in V$ соответствует непустое подмножество $D_i \subseteq D$. Существование ребра $v_i \to v_j$ между v_i и v_j означает, что документы, принадлежащие D_i предшествуют документам из D_j. Иными словами, для понимания документов из D_j необходимо ознакомиться с документами из D_i.

Дерево чтения D — это порядок чтения $R(V, E)$ со следующими свойствами:

1. Для каждой вершины $v_i \in V$ с соответствующим множеством документов D_i справедливо: документ $d \in D$ соответствует вершине v_i тогда и только тогда, когда $d \leftrightarrow d_i$ для всех $d_i \in D_i$.

2. Для каждой пары вершин $v_i, v_j \in V$ с наборами документов D_i и D_j соответственно, для ребра $v_i \to v_j$ справедливо: для каждой пары документов $d_i \in D_i$ и $d_j \in D_j$ верно $d_i \to d_j$.

3. Для каждой пары документов $v_i, v_j \in V$ с наборами документов D_i и D_j соответственно, и существующим ребром $v_i \to v_j$ справедливо, что не существует другой вершины v_k, такой, что: $v_i \to v_k \to v_j$.

Цепочка чтения $d_{m1} \to d_{m2} \ldots \to d_{mk}$ документов $d_{mi} \in D, i = 1 \ldots k$, соответствует пути $v_{l1} \to v_{l2} \ldots \to v_{lk}$ в графе с $v_{li} \in V, i = 1 \ldots k$, так, что, $d_{mi} \in D_{li}$ для вершины $v_{li}, i = 1 \ldots k$.

Задача *Reading Order* состоит в построении дерева чтения по заданной коллекции документов. Мерой качества в данной задаче является средний квадрат ошибки между матрицами смежности дерева чтения, построенного алгоритмом, и эталонного дерева чтения, содержащее реальные цепочки чтения для данной коллекции.

Формально, *матрицей смежности* дерева D назовем такую матрицу A, что:

$$A_{ij} = \begin{cases} \frac{1}{E(d_i \to d_j)}, & \text{если существует путь } d_i \to d_j, \\ 0, & \text{иначе} \end{cases} \tag{16}$$

где $E(d_i \to d_j)$ — количество ребер на кратчайшем пути из d_i в d_j.

Тогда мера близости между деревьями чтения D и $\hat{D}$ с матрицами смежности A и $\hat{A}$ соответственно задается как средний квадрат ошибки (MSE):

$$MSE(A, \hat{A}) = \frac{1}{n} \sum_{i,j=1}^{n} (A_{ij} - \hat{A}_{ij})^2 \tag{17}$$

8.2 Описание алгоритма

Алгоритм 1, предложенный в [10], базируется на двух статистиках: оценке общности документа g и мере близости документов ρ. Обе оценки требуют построенной тематической модели, которая определяет для документов $d_1, d_2, \ldots d_n \in D$ тематические векторы $\theta_1, \theta_2, \ldots \theta_n$. Тогда, функция общности документа d $g(d)$ задается энтропией тематического вектора θ документа d:

$$g(d) = -\sum_{j=0}^{T} \theta_j \log \theta_j, \tag{18}$$

где T — размерность тематического вектора (количество тем в тематической модели).

Функция $\rho(d_1, d_2)$ задает меру близости между документами d_1 и d_2. К примеру, в качестве $\rho(d_1, d_2)$ может быть взято косинусное расстояние между тематическими векторами документов d_1 и d_2, описанное в разделе 5.

Документы d_1, d_2 считаются эквивалентными (принадлежащими одной вершине, $d_1 \leftrightarrow d_2$), если $g(d_1) - g(d_2) < \tau$ и $\rho(d_1, d_2) > k$, где τ и k — гиперпараметры модели.

Работа [10] предлагает и доказывает корректность алгоритма 1. Алгоритм жадно формирует вершину v из эквивалентных документов, затем отбирает все достаточно близкие документы ко всем документам вершины из оставшихся. Затем производится кластеризация полученного набора документов, и каждый полученный кластер образует множество документов поддерева. Затем процедура рекурсивно повторяется для каждого кластера, формируя поддеревья вершины v. Алгоритм останавливается, когда заканчиваются документы, не принадлежащие ни одной вершине.

Согласно алгоритму, в общем случае, получается *лес деревьев чтения*, каждое из которых соответствует своей тематике.

функция ГенерацияДерева:
- **Вход:** набор документов D, матрица тематических векторов Θ, порог общности τ, порог близости k, текущая вершина v_r
- **Выход:** Структура дерева
- **пока** $D \neq 0$
  - Создать множество S, содержащее самый общий документ $d \in D$
  - Выбрать следующий самый общий d_j in D
  - **пока** $\rho(d, d_j) > k$ и $g(d) - g(d_j) < \tau$
    - **если** $\rho(d_i, d_j) > k \; \forall d_i \in S$ **то**
      - Добавить d_j в S
    - Выбрать следующий самый общий d_j in D
  - `/* S содержит самые общие эквивалентные документы и соответствует новой вершине` v_s `*/`
  - Добавить ребро $v_r \to v_s$
  - $D = D \setminus S$
  - $C \leftarrow \{d_j \in D | \rho(d, d_j) > 0\}$
  - Разделить C на кластеры D_c т.ч.:
  - для всех кластеров D_c справедливо $\rho(d_i, d_j) > 0, \forall d_i, d_j \in D_c$
  - **для всех** кластеров D_c
    - Генерация Дерева$(D_c, \Theta, \tau, k, v_s)$

Алгоритм 1. Алгоритм генерации дерева чтения [10].

8.3 Предложенная модификация

Когнитивные оценки общности могут улучшить алгоритм на этапе формирования множества документов вершины. Оценки общности хоть и показывают насколько широко документ покрывает темы коллекции, однако, из этого нельзя сделать вывод, что он содержит более простой контент.

К примеру, используя построенную в разделе 5 тематическую модель, можно обнаружить, что у статьи «Оптимизация» значение энтропии равно 0.87, а у статьи «Градиентный спуск» — 1.35. Это связано с тем, что в первом документе говорится преимущественно про простую математику, в то время как в статье про градиентный спуск упоминаются свойства из математического анализа, теории оптимизации, описываются приложения к задачам машинного обучения и нейронным сетям. Покрываемый спектр тем последней статьей действительно шире, но по логике статья «Оптимизация» должна быть прочитана раньше.

Оценки когнитивной сложности помогают скорректировать оценки общности. Пусть $f(\Delta_{d_1,d_2}g, \Delta_{d_1,d_2}W)$ — функция от разностей значений общности и какой-либо (из описанных выше) оценки сложности между документами d_2 и d_1 ($\Delta_{a,b}g = g(b) - g(a)$). Предлагаемая модификация алгоритма 1 состоит в замете условия $g(d) - g(d_j) < \tau$ в процессе формирования вершины, на $f(\Delta_{d_j,d}g, \Delta_{d_j,d}W) < \tau$. Обновленный алгоритм имеет вид 2.

Эксперименты и значения метрики качества для различных видов функции f и оценок сложности W обсуждаются в подразделе «Результаты».

функция ГенерацияДерева:

Вход: набор документов D, матрица тематических векторов Θ, порог общности τ, порог близости k, текущая вершина v_r

Выход: Структура дерева

пока $D \neq 0$

Создать множество S, содержащее самый общий документ $d \in D$

Выбрать следующий самый общий d_j in D

пока $\rho(d, d_j) > k$ и $f(g(d) - g(d_j), W(d) - W(d_j)) < \tau$

если $\rho(d_i, d_j) > k \; \forall d_i \in S$ **то**

Добавить d_j в S

Выбрать следующий самый общий d_j in D

Добавить ребро $v_r \to v_s$

$D = D \setminus S$

$C \leftarrow \{d_j \in D | \rho(d, d_j) > 0\}$

Разделить C на кластеры D_c т.ч.:

для всех кластеров D_c справедливо $\rho(d_i, d_j) > 0, \forall d_i, d_j \in D_c$

для всех кластеров D_c

Генерация Дерева$(D_c, \Theta, \tau, k, v_s)$

Алгоритм 2. Модифицированный алгоритм генерации дерева чтения.

8.4 Результаты

В качестве набора данных были взяты статьи русскоязычной Википедии по теме машинного обучения. Для экспериментов использовались тематические векторы, полученные из тематической модели ARTM, описанной в разделе 5. Для валидации в качестве эталонного дерева была взята иерархия категорий Википедии, доступная на сайте wikimedia.org. Структура иерархии категории Википедии полностью соответствует определению дерева чтения. Во всех экспериментах $k = 0.8$, значения τ указаны в таблице. Результаты представлены в таблице 5. Пример дерева, построенного лучшей моделью, приведен на рис. 8.

Функция f	**Оценка сложности**	**MSE**
$f(\Delta g, \Delta W) = \Delta g, \tau = 0.1$	—	0.153
$f(\Delta g, \Delta W) = \Delta W, \tau = 0.1$	Морфологическая (слоги)	0.182
	Лексическая (частотная)	0.178
	Агрегированная	0.178
$f(\Delta g, \Delta W) = \frac{\Delta g + \Delta W}{2}, \tau = 0.1$	Морфологическая (слоги)	0.130
	Лексическая (частотная)	0.159
	Агрегированная	**0.124**
$f(\Delta g, \Delta W) = \frac{\Delta g \Delta W}{\Delta g + \Delta W}, \tau = 0.1$	Морфологическая (слоги)	0.141
	Лексическая (частотная)	0.149
	Агрегированная	0.138

Таблица 5: Сравнение оригинального и модификационного алгоритма построения дерева чтения в зависимости от вида функции f и типа оценок сложности.

Используя среднее арифметическое оценок общности и агрегированной сложности, удается улучшить результат. Среднее арифметическое отлично от нуля, если одна из его переменных отлична от нуля, в отличие от среднего гармонического. Это позволяет блокировать добавление документа в вершину, если он имеет более высокую сложность, чем остальные документы вершины. Использование только оценок сложности без поправки на общность работает хуже. Так, рождается гипотеза, что сложность и общность — лишь некоторые характеристики документа, объединяя которые, можно получить продвижение в задаче Reading Order. Поискам таких характеристик документов будут посвящены дальнейшие исследования.

Рис. 8: Пример одного из деревьев чтения для коллекции Википедии .

9 Ресурс TextComplexity.net

Для демонстрации работы алгоритмов был разработан интерактивный веб-ресурс *TextComplexity.net*. Интерфейс позволяет использовать модели оценивания сложности на фонетическом, морфологическом, лексическом и синтаксическом уровнях. Вводя произвольный текст в текстовое поле, система подсвечивает аномально сложные токены на выбранном уровне. Интерфейс системы показан на рис. 9.

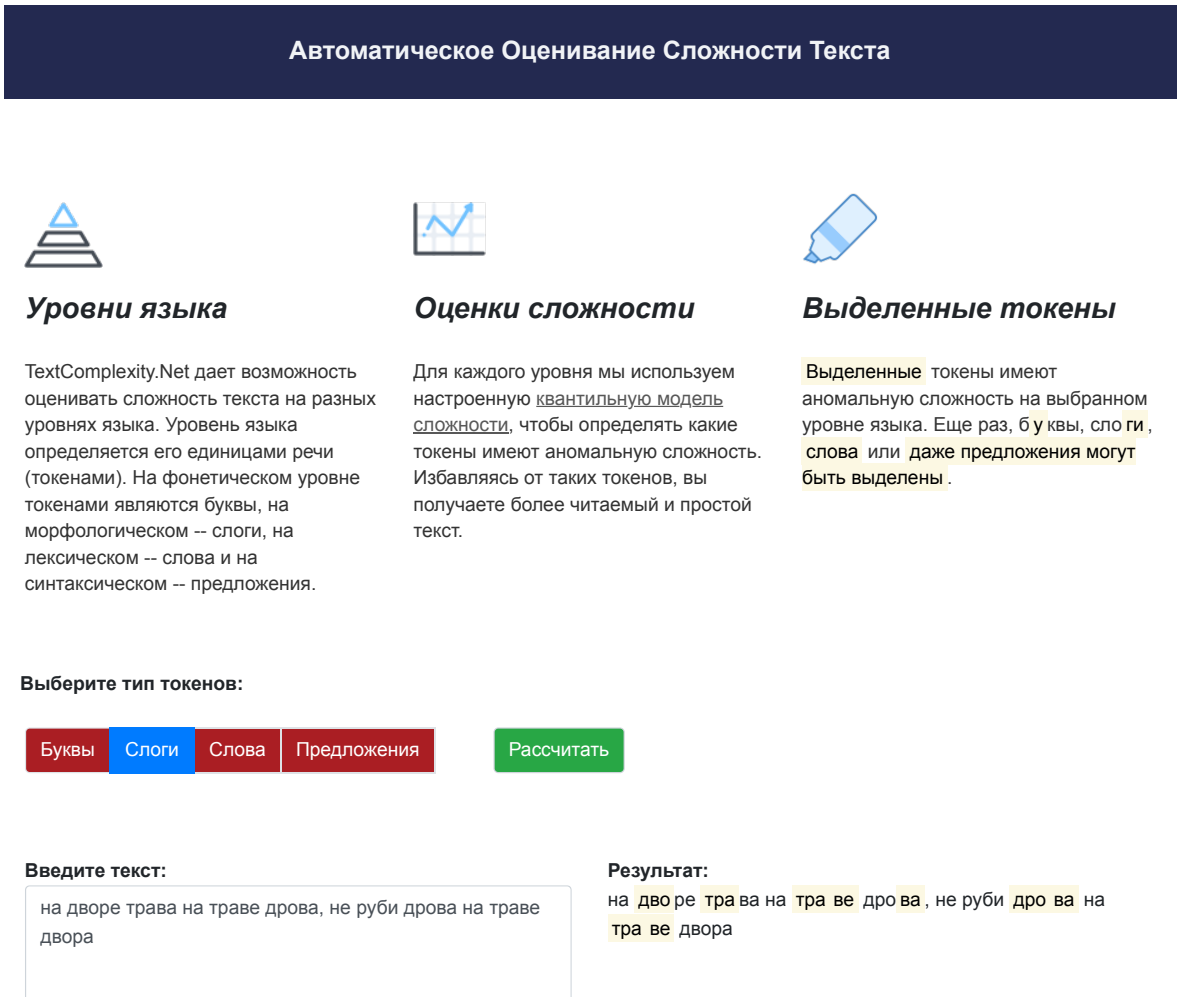


Рис. 9: Интерфейс системы TextComplexity.net.

Система полезна в редакторских целях. Благодаря выделению аномально сложных токенов в режиме онлайн, можно быстро и эффективно редактировать текст с целью уменьшения количества сложных токенов на каждом уровне. Пример оригинального и отредактированного текста с применением системы изображен на рис. 10.

Очевидно, что при крайне большом наборе обучающих данных алгоритм обучения нейронной сети будет работать крайне медленно, поэтому на практике часто проводят модификацию коэффициентов сети после каждого элемента обучения, где значения градиента целевой функции приближаются градиентом функции стоимости, вычисленном только на одном элементе обучающей выборки. Такой тип градиентного спуска называется оперативный градиентный спуск или стохастический градиентный спуск. Стохастический градиентный спуск является одной из форм стохастического приближения. Теория стохастических приближений доказывает условия сходимости стохастического градиентного спуска.
На большом наборе данных алгоритм обучения будет работать медленно, поэтому на практике часто корректируют коэффициенты сети после каждого объекта, где значение градиента аппроксимируются градиентом функции стоимости, вычисленном только на одном элементе выборки. Такой метод называют стохастическим градиентным спуском. Теория стохастических приближений дает условия сходимости описанного метода.

Рис. 10: Пример оригинального текста (выше) и исправленного текста (ниже). Красным цветом выделены аномально сложные токены на лексическом уровне.

Библиотека с открытым кодом. Ресурс разработан на базе собственной библиотеки с открытым кодом *Cognitive Complexity* (https://github.com/maks5507/cognitive-complexity). Библиотека позволяет обучать модели на представленных уровнях языка в параллельном режиме, обучать агрегированные оценки сложности, используя выборку размеченных пар документов, а также визуализировать полученные эмпирические распределения.

10 Заключение

В данной работе исследован метод оценивания когнитивной сложности текста на основе психофизиологических предположений. Расширены методы, предложенные в [3, 2, 24]. В частности, разработаны модели на фонетическом, морфологическом, лексическом, синтаксическом и дискурсивном уровне языка, а также построена агрегированная модели сложности текста, комбинирующая модели со всех уровней. Все модели сложности обучаются по референтному корпусу несложных текстов, считая эмпирические распределения значений сложностей каждого токена отдельно или всех сразу. Предложен подход к оцениванию качества подобных моделей, использующий собранный путем краудсорсинга набор данных. По результатам экспериментов,

проведенных на двух референтных корпусах различной природы, все предложенные модели демонстрируют более высокие результаты чем индексы удобочитаемости и ранее используемые модели. Описанный подход применен к задаче построения деревьев чтения, улучшив требуемые метрики качества. Наконец, реализован веб-ресурс TextComplexity.net, позволяющий в интерактивном режиме определять аномально сложные токены в произвольном тексте вместе с библиотекой с открытым кодом Cognitive Complexity.

Промежуточные результаты, описанные в данной работе, были представлены на конференциях *Recent Advances in Natural Language Processing*, *Математические Методы Распознования Образов*, *DataFest*, *OpenTalks.AI*. В частности, статья [6] описывает эксперименты с лексическими оценками сложности, а последние результаты будут представлены на конференции Диалог-2020.

Данная работа является шагом к полноценному решению задачи Reading Order. Оценки сложности, наряду с оценками общности, составляют часть численных характеристик документов, которые можно эффективно применять для ранжирования результатов разведочного поиска в логическом порядке чтения. Данное исследование планируется продолжить в этом направлении.

Список литературы

[1] Bigartm: Open source library for regularized multimodal topic modeling of large collections / Konstantin Vorontsov, Oleksandr Frei, Murat Apishev et al. // AIST'2015, Analysis of Images, Social networks and Texts. — Springer International Publishing Switzerland, Communications in Computer and Information Science (CCIS), 2015. — Pp. 370–384.

[2] *Birkin A.A.* Speech Codes. — Saint-Peterburg: Hippocrat, 2007.

[3] *Birkin A.A.* Nature of Speech. — Moscow: Likbez, 2009.

[4] Coh-metrix: Analysis of text on cohesion and language / Arthur Graesser, Danielle McNamara, Max Louwerse, Zhiqiang Cai // *Behavior research methods, instruments and computers : a journal of the Psychonomic Society, Inc.* — 2004. — 06. — Vol. 36. — Pp. 193–202.

[5] *Dzmitryieva Aryna.* The art of legal writing: A quantitative analysis of russian constitutional court rulings // *Sravnitel'noe konstitucionnoe obozrenie.* — 2017. — 01. — Vol. 3. — Pp. 125–133.

[6] *Eremeev M. A., Vorontsov Konstantin.* Lexical quantile-based text complexity measure // RANLP. — 2019.

[7] *Flesh R.* How to test readability // *New York, Harper and Brothers.* — 1951.

[8] *Gunning Robert.* The technique of clear writing. — New York: McGraw-Hill, 1952.

[9] *Hofmann Thomas.* Probabilistic latent semantic indexing // Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. — SIGIR '99. — New York, NY, USA: ACM, 1999. — Pp. 50–57.

[10] *Koutrika Georgia, Liu Lei, Simske Steven.* Generating reading orders over document collections // 2015 IEEE 31st International Conference on Data Engineering. — 2015. — April. — Pp. 507–518.

[11] *Marchionini Gary.* Exploratory search: From finding to understanding // *Commun. ACM.* — 2006. — Vol. 49, no. 4. — Pp. 41–46.

[12] *McLaughlin G. H.* Smog grading: A new readability formula // *Journal of Reading.* — 1969. — Vol. 12(8). — Pp. 639–646.

[13] Non-bayesian additive regularization for multimodal topic modeling of large collections / Konstantin Vorontsov, Oleksandr Frei, Murat Apishev et al. — 2015. — 10.

[14] *Oborneva Irina.* Avtomatizirovannaya otsenka slozhnosti uchebnykh tekstov na osnove statisticheskikh parametrov [semiautomatic evaluation of the complexity of academic texts on the base of statistic parameters]. — 2006.

[15] Readability assessment for text simplification / Sandra Aluisio, Lucia Specia, Caroline Gasperin, Carolina Scarton. — 2010. — 06.

[16] Scalable topical phrase mining from text corpora / Ahmed El-Kishky, Yanglei Song, Chi Wang et al. // *Proceedings of the VLDB Endowment.* — 2014. — 06. — Vol. 8.

[17] *Senter R.J., Smith E.A.* Automated readability index // *AMRL-TR.* — 1967. — Vol. 66, no. 22.

[18] *Solovyev Valery, Ivanov Vladimir, Solnyshkina Marina.* Assessment of reading difficulty levels in russian academic texts: Approaches and metrics // *Journal of Intelligent and Fuzzy Systems.* — 2018. — 04. — Vol. 34. — Pp. 1–10.

[19] *Straka Milan, Strakova Jana.* Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. — 2017. — 01. — Pp. 88–99.

[20] A survey of definitions and models of exploratory search / Emilie Palagi, Fabien Gandon, Alain Giboin, Raphaël Troncy // ESIDAT17 - ACM Workshop on Exploratory Search and Interactive Data Analytics, Mar 2017, Limassol, Cyprus. — 2017. — 03. — Pp. 3–8.

[21] *Vorontsov K. V., Potapenko A. A.* Additive regularization of topic models // *Machine Learning, Special Issue on Data Analysis and Intelligent Optimization with Applications.* — 2015. — Vol. 101, no. 1. — Pp. 303–323.

[22] *White Ryen W., Roth Resa A.* Exploratory Search: Beyond the Query-Response Paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services. — Morgan and Claypool Publishers, 2009.

[23] *Zou Hui, Hastie Trevor.* Regularization and variable selection via the elastic net // *Journal of the Royal Statistical Society, Series B.* — 2005. — Vol. 67. — Pp. 301–320.

[24] *Тютюнник В. М., Биркин А. А., Гущин Ю. Г.* Основы лингвистической психофизиологии. — 2016. — 192 pp.

A Примеры эмпирических распределений

A.1 Фонетический уровень

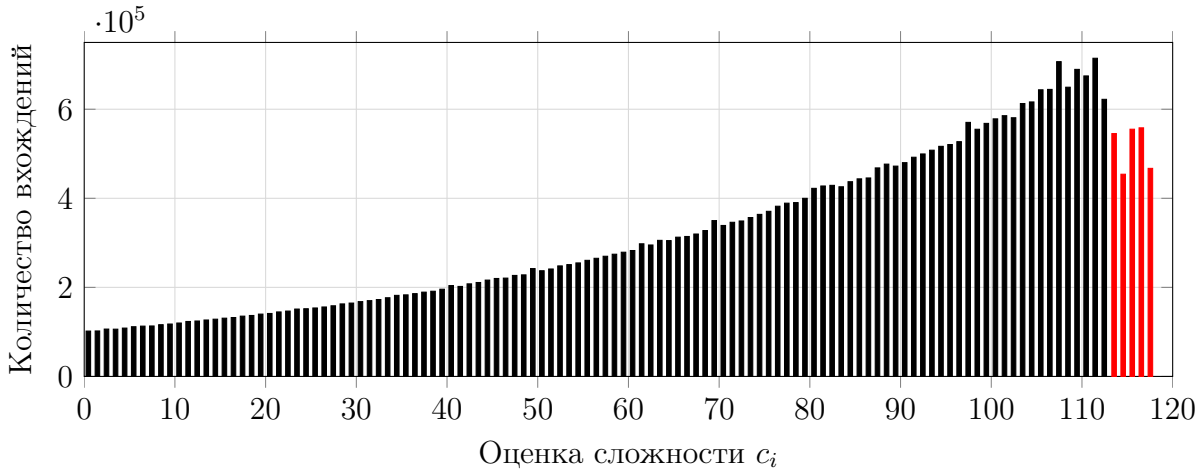


Рис. 11: Распределение оценок фонетической сложности c_i для буквы «У», посчитанное по русскоязычной Википедии. Красный хвост распределения соответствует $c_i > C_\gamma(x_i), \gamma = 0.95, \beta = 0.6$.

A.2 Морфологический уровень

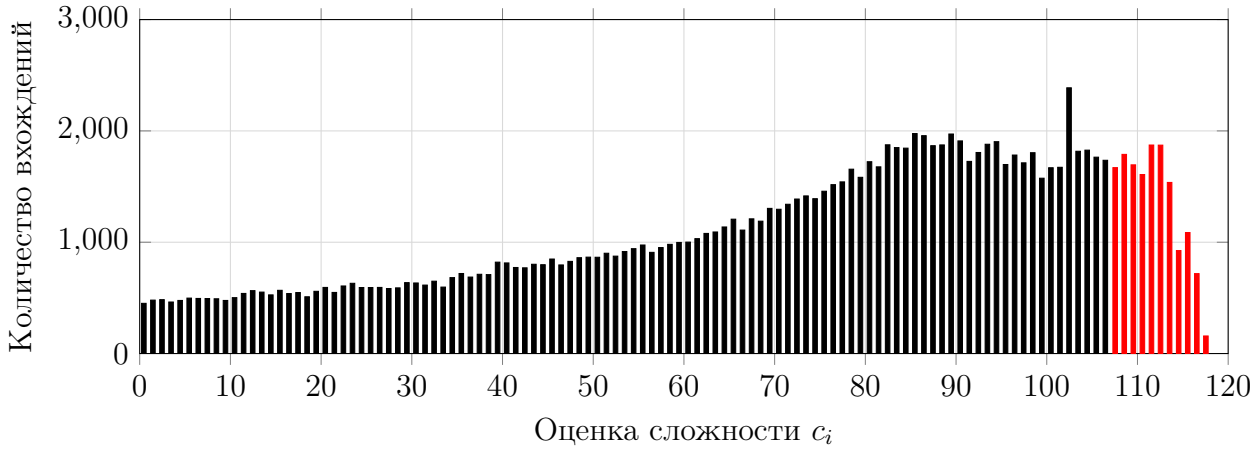


Рис. 12: Распределение оценок морфологической сложности c_i для слога «ЛОК» (с учетом порядка букв), посчитанное по русскоязычной Википедии. Красный хвост распределения соответствует $c_i > C_\gamma(x_i), \gamma = 0.95, \beta = 0.6$.

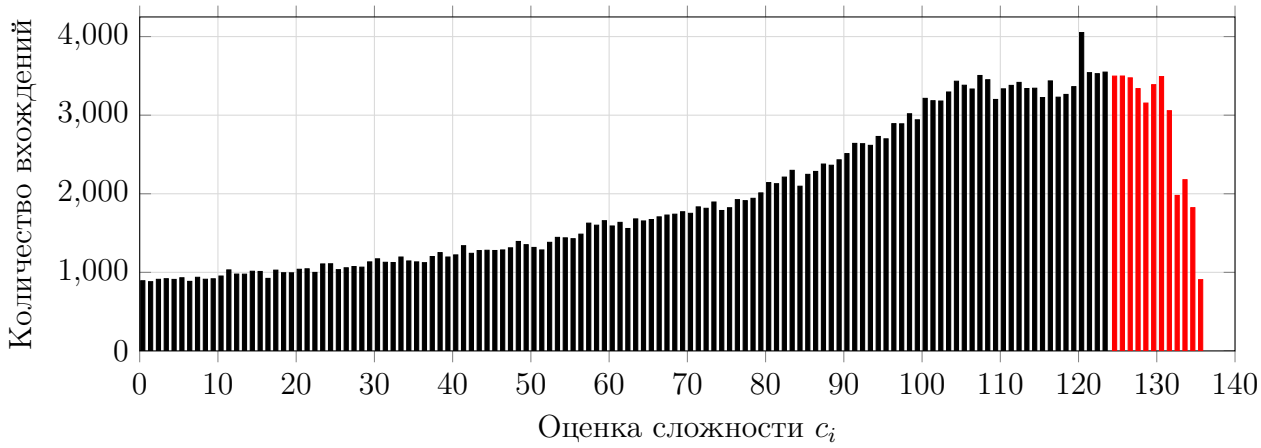


Рис. 13: Распределение оценок морфологической сложности c_i для слога «КЛО» (без учета порядка букв), посчитанное по русскоязычной Википедии. Красный хвост распределения соответствует $c_i > C_\gamma(x_i), \gamma = 0.95, \beta = 0.6$.

A.3 Лексический уровень

Рис. 14: Распределение оценок лексической сложности c_i для слова «МАТЕМАТИКА», посчитанное по русскоязычной Википедии. Красный хвост распределения соответствует $c_i > C_\gamma(x_i), \gamma = 0.95, \beta = 0.6$.

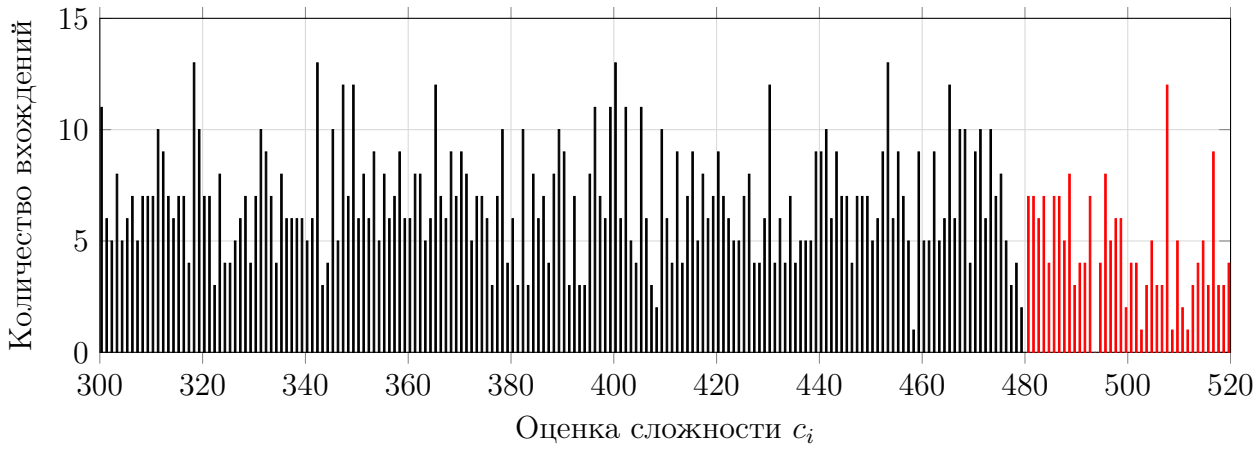


Рис. 15: Распределение оценок лексической сложности c_i для слова «МАТЕМАТИКА», посчитанное по коллекции документов «Ноосфера». Красный хвост распределения соответствует $c_i > C_\gamma(x_i), \gamma = 0.95, \beta = 0.6$.

A.4 Синтаксический уровень

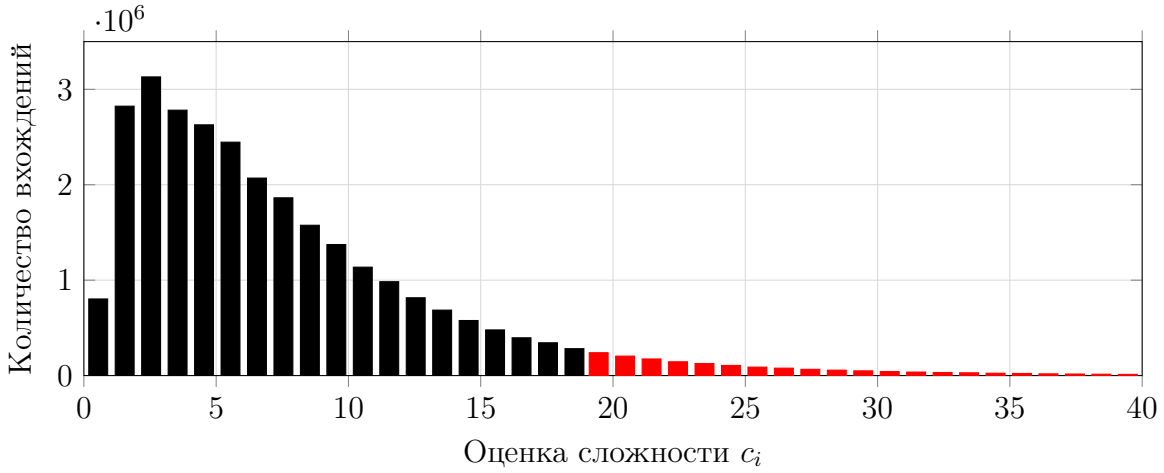


Рис. 16: Распределение сложностных оценок синтаксической сложности c_i (длина синтаксической связи), посчитанное по русскоязычной Википедии. Красный хвост распределения соответствует $c_i > C_\gamma, \gamma = 0.95$.

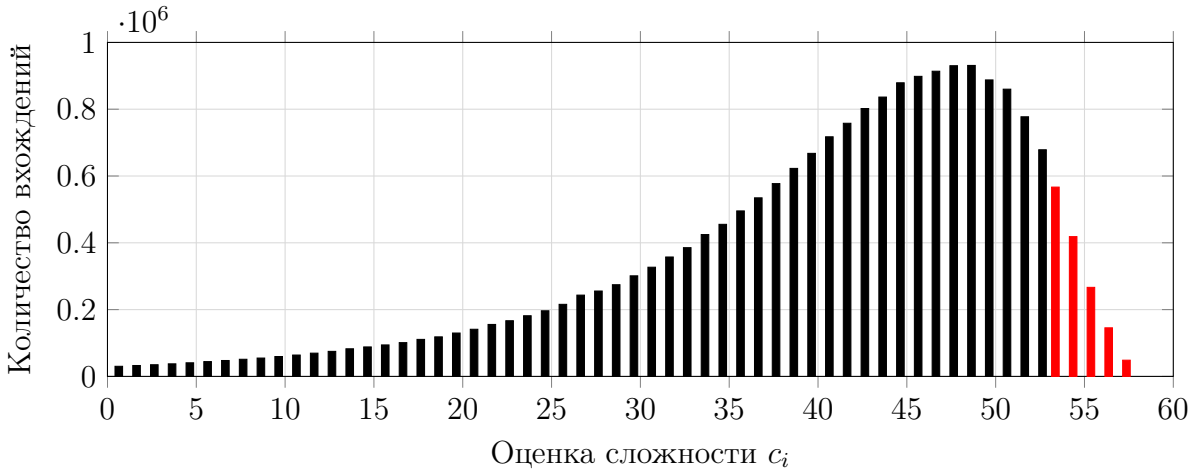


Рис. 17: Распределение частотных оценок синтаксической сложности c_i, посчитанное для токена (VERB, ROOT) по русскоязычной Википедии. Красный хвост распределения соответствует $c_i > C_\gamma(x_i), \gamma = 0.95, \beta = 0.6$.