УДК 004.912

Я. С. Пикалёв[1], Т. В. Ермоленко[2]
[1]Государственное учреждение «Институт проблем искусственного интеллекта», г. Донецк
83048, г. Донецк, ул. Артема, 118-б
[2]Государственное образовательное учреждение высшего профессионального образования «Донецкий национальный университет», г. Донецк
83000, г. Донецк, пр. Театральный, 13

СИСТЕМА АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ТРАНСКРИПЦИЙ РУССКОЯЗЫЧНЫХ СЛОВ-ИСКЛЮЧЕНИЙ НА ОСНОВЕ ГЛУБОКОГО ОБУЧЕНИЯ

Ya. S. Pikalyov[1], T. V. Yermolenko[2]
[1]Public institution «Institute of Problems of Artificial Intelligence», Donetsk
83048, Donetsk, Artema str., 118-b
[2]State Educational Institution of Higher Professional Education «Donetsk National University»
83000, Donetsk, Teatralnyiy av., 13

SYSTEM OF AUTOMATIC TRANSCRIPTION GENERATION OF RUSSIAN-LANGUAGE WORDS EXCEPTIONS ON THE BASIS OF DEEP LEARNING

В статье рассмотрены основные особенности фонетики и орфоэпии русского языка, которые необходимо учитывать при генерации транскрипции, приведено краткое описание современных подходов получения транскрипции. Особое внимание уделено нейросетевым архитектурам, использующимся для задачи графемно-фонемного выравнивания. Для автоматической генерации транскрипции слов-исключений предложен метод кодирования слов, а также нейросетевая модель на основе архитектуры Transformer, модифицированной с помощью техники «teacher forcing», градиентного отсечения, а также механизма RL-block, в котором реализовано совместное применение обучения с учителем и обучения с подкреплением. Предложенная модификация позволила повысить точность модели генерации транскрипций для слов-исключений по критерию PER на 9%, по критерию WER – на 3%.
Ключевые слова: автоматическая генерация транскрипций; модель seq2seq, модель с вниманием, архитектура Transformer, RL-block.

The article describes the main features of phonetics and orthoepy of the Russian language, which must be considered when generating transcription, provides a brief description of modern approaches to obtaining transcription. Special attention is given to the neural network architectures used for the grapheme-phonematic alignment problems. For automatic generation of transcription of word-exceptions, the method of word encoding is proposed, as well as a neural network model based on the Transformer architecture, modified by the "teacher forcing" technique, gradient clipping, as well as the RL-block mechanism, which implements the joint use of teacher training and learning with the reinforcement. The proposed modification made it possible to increase the accuracy of the transcription generation model for PER-exclusion words by 9%, and by 3% according to WER criterion.
Key words: automatic generation of transcriptions; seq2seq model, attention model, Transformer architecture, RL-block.

Введение

Для реализации системы распознавания слитной речи единицы распознавания должны быть связаны с единицами фонетического уровня. Поэтому вместо создания моделей для каждого слова создаются модели элементов нижнего уровня (слоги, фонемы и т.п.). Необходимость использовать части фонем и контекстную зависимость объясняется коартикуляцией (взаимным влиянием произносимых звуков друг на друга), ассимиляцией (объединения звуков), а также редуцированием (сокращения длительности определенных звуков вплоть до полного исчезновения). Причем в разговорном стиле речи эти явления могут возникать как внутри слова, так и на стыках слов. Это приводит к значительному снижению точности автоматического распознавания речи.

В настоящее время общепринятым является использование контекстно-независимых фонем (монофонов) для средних словарей и контекстно-зависимых фонем (дифонов, трифонов) для больших словарей. Возникает необходимость создания словаря, содержащего слова с их орфографическим и фонетическим представлением, который обычно создается с использованием канонических фонетических правил транскрибирования для определенного языка. При этом именно генерация транскрипции слов является одним из важных шагов.

Для генерации транскрипции слов на русском языке достаточно знать позицию ударения и фонетические правила. Алгоритм получения транскрипции для слов-исключений, которыми изобилует современный русский язык, не подчиняется правилам фонетики и орфоэпии русского языка. В связи с этим актуальной задачей является построение моделей автоматической генерации транскрипций для слов-исключений, учитывающих влияние позиции ударения в слове (заударные, предударные гласные, побочные ударения и т.п.), а также явления коартикуляции, редукции и ассимиляции звуков русской речи.

Особенности фонетики русского языка

При разработке системы формирования автоматической транскрипции должен быть предусмотрен ряд дополнений, учитывающих особенности фонетики русского языка. Эти дополнения были сформированы на основе информации, полученной из работ [1-6].

1. **Слова-исключения.** Под понятием «слово-исключение» в данной работе подразумевается слово, не подчиняющееся правилам фонетики и орфоэпии русского языка для получения транскрипции. Большая часть слов-исключений являются иноязычными словами. В силу длительных экономических, политических, культурных, военных и иных связей русского народа с другими в его язык проникло значительное количество иноязычных слов, которые имеют различную степень ассимиляции и неограниченную или ограниченную сферу употребления. В русской лексикологической традиции выделяются: слова, давно усвоенные и используемые наравне с русскими («стул», «лампа», «школа» и т.д.); слова, не всем понятные, но необходимые, так как они обозначают понятия науки, техники, культуры и т.п. («фонема», «морфема», «дагностицизм» и т.п.); слова, которые могут быть заменены исконно русскими без всякого ущерба для смысла и выразительности высказывания («эпатировать», «эпатаж», «апологет», «акцентировать», «визуальный» и т.п.). Сейчас значительная часть таких слов по своему произношению ничем не отличается от слов исконно русских. Но некоторые из них – слова из разных областей техники, науки, культуры, политики. Иноязычные собственные имена выделяются среди других слов русского литературного языка своим произношением, как правило, не следуя фонетическим и орфоэпическим нормам русского языка.

2. Кроме иноязычных слов к словам-исключениям относят слова, чьё произношение предопределяется литературной или диалектической нормой. Например, «что» произносится как [што], потому что это соответствует литературной норме (большинство производных от слова «что» тоже произносится с использованием звукосочетания [шт]: «что-либо», «что-нибудь» и т.п.).

3. **Сложносоставные слова**. Большую трудность вызывают сложносоставные слова – слова, состоящие из двух и более основ. Эта трудность вызвана наличием более одного ударения в слове, в связи с чем стандартные правила транскрипции не применимы. Явления, когда в образовании сложного слова используется более двух корней, достаточно редки («веломотодром»). Следует отличать сложные слова от простых. Так, в слове «электрификация» всего один корень «электри-», а все, стоящее за ним, – это суффикс и окончание.

4. **Предударные гласные (слоги).** Ударения для гласных осуществляются следующим образом: алгоритмом определяется позиция ударения; все гласные, стоящие перед ударной гласной, являются предударными гласными (слогами) и обозначаются как «А_» (самая ближняя предударная обозначается как «А_» (где А – звук «а»), в свою очередь остальные предударные обозначаются, например, как «А__». Стоит отметить, что во втором и третьем предударных слогах гласные подвергаются более значительной редукции, чем в первом слоге.

5. **Заударные гласные (слоги).** Гласные, стоящие после ударной гласной, являются заударными и обозначаются как «А*», в свою очередь самая ближняя обозначается как «А*», а самая дальняя как «А**». Произношение гласных в заударных слогах, в большинстве случаев, аналогично произношению гласных во всех предударных слогах, кроме первого.

6. **Побочное ударение.** Многие сложносоставные слова (имеющие более одного корня) кроме основного ударения могут иметь побочное (или побочные). При наличии двух ударений в слове побочным, как правило, объявляется ударение, находящееся ближе к началу слова, а основным ударением объявляется ударение, находящееся ближе к концу слова. Побочное ударение характеризует свободный стиль речи («общежитие», «девятьсот»). Помимо сложносоставных слов, побочное ударение могут иметь и сложносокращённые слова («Донгормаш»). Также побочное ударение могут иметь приставки в словах («чрезмерный»). С побочным ударением обычно произносятся слова иноязычного происхождения («постскриптум»). Если в сложносоставном слове три основы, то оно может иметь три ударения – 2 побочных и 1 основное («авиаметеослужба»).

7. **Слова с апострофом**. Апостроф является так называемым небуквенным орфографическом знаком. При этом в ряде слов апостроф логически делит слово на подслова («д' Ареццо» = «д» + «Ареццо»), также может входить в фонетическую основу слова (Word'a = «ворда»).

Подходы к автоматической генерации транскрипции слов

Существующие подходы фонемного транскрибирования, реализованные в современных системах распознавания речи, можно разбить на два направления:

1) на основе знаний («традиционный» подход);

2) на основе данных (статистический подход).

Методы «традиционного» подхода используют словарь или набор лингвистических правил [7-9], сформированные экспертом-лингвистом. Методы статистиче-

ского подхода [10], [11] заключаются в обучении алгоритма транскрибирования по словарю, содержащем буквенные и фонемные формы представления слов. Недостаток подхода на основе знаний заключается в ограниченности словаря и необходимости ручного составления набора правил. Недостатком подхода на основе данных является зависимость результата от обучающих данных.

Генерации транскрипций слов посвящён ряд работ [12-16], в работах [17-19] рассматриваются методы генерации транскрипций для русского языка. Как правило, разработчики систем построения транскрипций в качестве формата представления транскрипций используют формат международного фонетического алфавита (International Phonetic Alphabet, IPA) [20].

Рассмотренные в указанных работах (общедоступные) системы автоматического формирования транскрипций не учитывают всех особенностей русского языка, а именно:

- степень предударности и заударности в гласных, т.к. она влияет на произношение;
- побочные ударения;
- произношение слов-исключений и слов с апострофами.

Также стоит отметить для «традиционного» подхода в случае, если слова нет в словаре транскрипций – невозможно точно сгенерировать транскрипцию. Поэтому разработка системы, объединяющей оба подхода, является актуальной задачей. То есть использовать подход, при котором используется унифицированная транскрипция – использовать словарь для получения транскрипции, а в том случае, если слова нет в словаре – использовать вероятностную модель для генерации транскрипции.

Среди алгоритмов генерации транскрипций слов, относящихся к группе статистического моделирования, наилучшие результаты показывает нейросетевой подход, подразумевающий наличие извлечённых из набора обучающих данных статистических зависимостей. В качестве обучающих данных обычно выступает словарь слов с их фонемными транскрипциями. На основе обучающего словаря происходит сопоставление букв с фонемами одного слова (задача графемно-фонемного выравнивания). Выделяют следующие виды сопоставлений между буквами и фонемами:

- один к одному (one-to-one): [самолёт] − [saymalljot];
- один ко многим (one-to-many): [самолёт] − [s][ay][m][a][ll][jo][t];
- многие ко многим (many-to-many): [с][а][м][о][л][ё][т] − [s][ay][m][a][ll][jo][t].

В качестве одной из нейросетевых архитектур, использующихся для задач обработки естественного языка, к которым относится и графемно-фонемное выравнивание, выделяют модель seq2seq (sequence-to-sequence, множество во множество) [21], базирующуюся на архитектуре RNN. Seq2seq (рис. 1) состоит из двух RNN: одна представляет собой енкодер (для обработки входных данных), а другая – декодер (для генерации выходного значения). Енкодер преобразует входную последовательность данных X в свое непрерывное представление Y, которое, в свою очередь, используется декодером для генерации вывода, по одному символу за раз.

Конечным состоянием кодировщика является вектор фиксированного размера z, который должен кодировать входную последовательность, используя предобученную модель. Это конечное состояние называется вложением последовательности (embedding) или контекстным вектором. Декодер использует полученный контекстный вектор для генерации выходных данных. Следовательно, формула для скрытых состояний енкодера имеет следующий вид:

$$h'_t = f(h'_{t-1,} y_{t-1}, c), \quad (1)$$

где y – итоговая последовательность; c – контекстный вектор:

$$c = f(x_1, x_2, ..., x_m), \quad (2)$$

где x – входная последовательность.

Декодер имеет следующий вид:

$$y_t = g(c, y_1, y_2, ..., y_{t-1}). \quad (3)$$

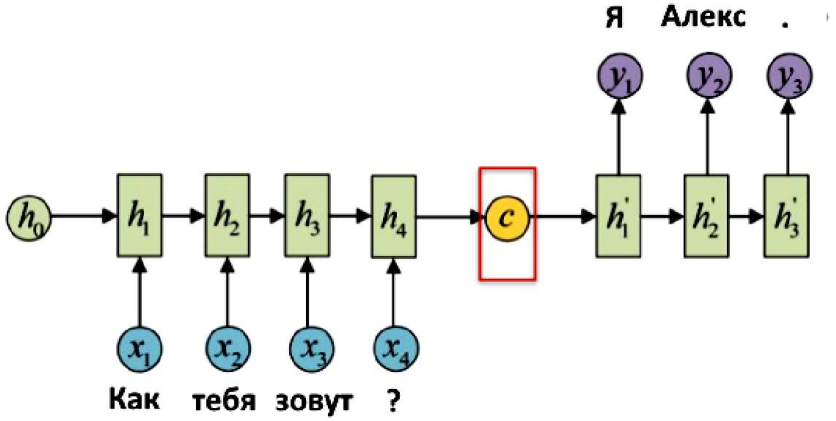

Рисунок 1 – Типовая схема работы seq2seq модели

В модели seq2seq исходная информация сжимается в вектор контекста фиксированной длины. Основным её недостатком является неспособность запоминать длинные предложения, что приводит к плохому результату. Эта проблема была решена за счёт использования модели с вниманием [22] (рис. 2). В данной модели вместо построения одного контекстного вектора из последнего скрытого состояния декодера создаётся контекстный вектор для каждого входного слова. Таким образом, если в исходном документе N уникальных слов, то должно быть создано N контекстных векторов, а не один. Преимущество применения данного подхода состоит в том, что закодированная информация хорошо декодируется моделью.

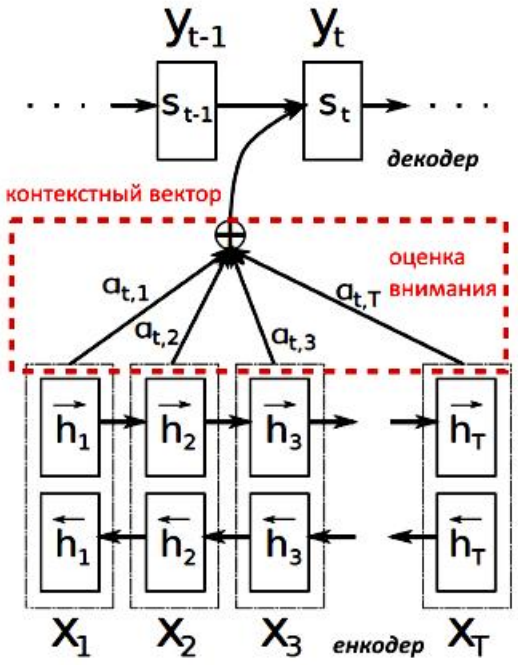

Рисунок 2 – Архитектура модели с вниманием

Формула контекстного вектора для модели внимания имеет вид:

$$c_t = \sum_{j=1}^{T_x} a_{tj} h_j , \tag{3}$$

где a_{tj} – веса для каждого скрытого состояния h_j (оценка внимания):

$$a_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{T_X} \exp(e_{tk})}, \tag{4}$$

где e_{tj} – модель вложения последовательности:

$$e_{tj} = a(s_{t-1}, h_j), \tag{5}$$

где s_t – скрытое состояние енкодера:

$$s_t = f(s_{t-1}, y_{t-1}, c_t). \tag{6}$$

Модель с вниманием имеет ряд недостатков: контекстный вектор вычисляется через скрытое состояние между исходной и целевой последовательностью, не учитывая контекст внутри исходного предложения и самого целевого предложения; данная модель из-за своей структуры сложна в распараллеливании.

Модель Transformer [23], [24] основана на seq2seq модели. Модель Transformer использует отдельные модели енкодера (преобразует слова входного предложения в один или больше векторов в определенном пространстве) и декодера (генерирует из этих векторов последовательность слов).

В качестве стандартных архитектур для енкодера и декодера Transformer использует полносвязные слои. Архитектура Transformer (рис. 3) нацелена на проблему трансдукции последовательности, что означает любую задачу, в которой входные последовательности преобразуются в выходные последовательности.

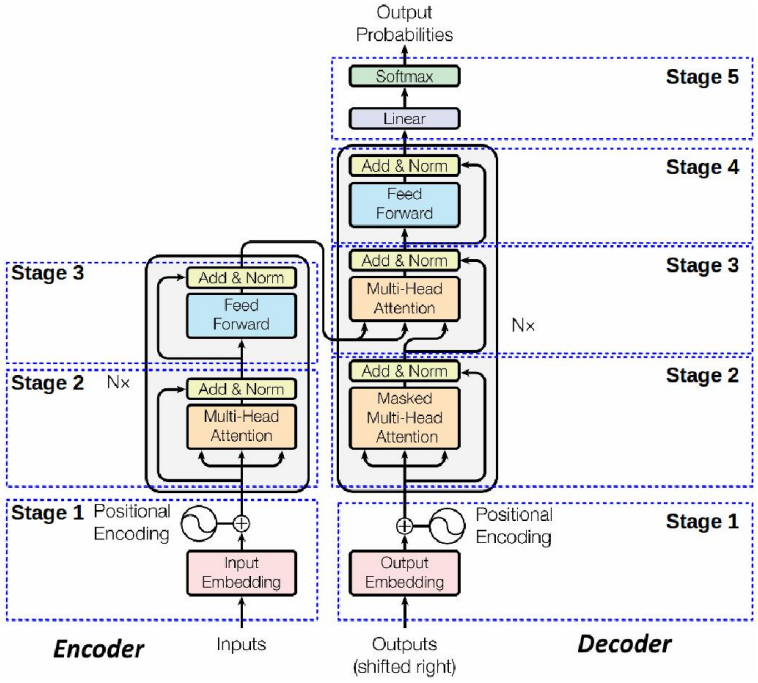

Рисунок 3 – Архитектура Transformer

Transformer уменьшает количество последовательных операций для привязки двух символов из последовательностей ввода-вывода с постоянным количеством операций O(1). Это достигается при помощи механизма многозадачности, который позволяет моделировать зависимости независимо от их расстояния во входном или выходном предложении. Новый подход Transformer заключается в том, чтобы полностью исключить повторение, и заменить его на обучающее внимание для обработки зависимостей между входом и выходом.

В Transformer енкодер и декодер состоят из стека одинаковых слоёв. Каждый из этих слоёв состоит из двух общих типов подслоёв:

- механизма многослойного обучающего внимания (multi-head attention);
- позиционной полносвязной нейросети прямого распространения (feed forward).

Главное отличие декодера от енкодера в Transformer – использование слоя с механизмом маскирующего многослойного внимания (masked multi-head attention), который позволяет «обращать внимание» на специфичные сегменты из енкодера (рис. 4). Это стало возможным благодаря механизму masked multi-head attention, который маскирует будущие токены посредством блокирования информации токенов, которые находятся справа от вычисляемой позиции.

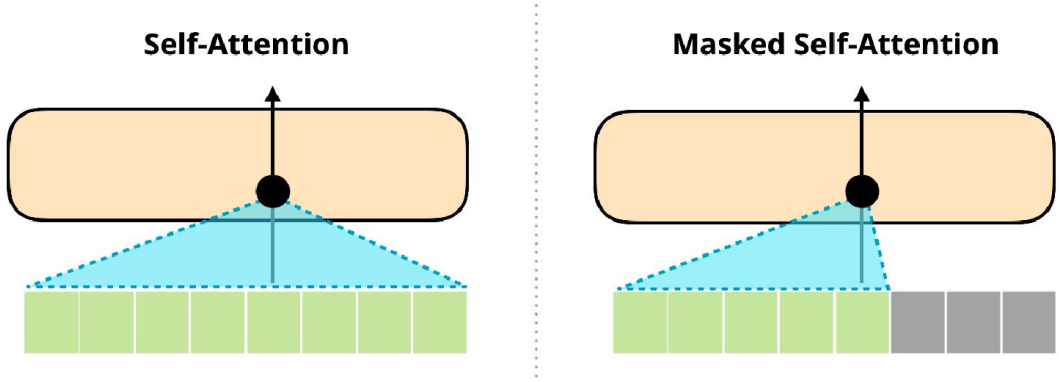

Рисунок 4 – Схема работы self-attention и masked self-attention

Модель генерации транскрипций для слов-исключений

Для обучения нейросети необходимо:

- создать словарь, содержащий слова и их фонемные транскрипции;
- преобразовать входные данные (слова) в вектор чисел.

В связи с этим был создан словарь, содержащий более 5 млн уникальных словоформ, а также метки «<» (начало слова), «>» (конец слова), «*» (токен для обозначения внесловарного символа). Кроме того, был подготовлен набор данных, составленный из найденных в Сети фонетических словарей слов-исключений, которые при помощи библиотеки pymorphy2 были дополнены парадигмами. В общей сложности количество элементов словаря слов-исключений составило около 10 тыс. пар.

Для преобразования слов в вектор был разработан алгоритм кодирования слов, который, в первую очередь, направлен на оптимизацию процесса обучения модели при помощи мини-пакетного типа обучения. При трансформации данных для мини-пакетного типа обучения стоит помнить об изменении длины фразы в массивах данных. Чтобы разместить фразы разных размеров в одном пакете, необходимо сделать матрицу E длины $L_{max} \times b_s$ (L_{max} – максимальное кол-во слов во фразе, b_s – размер пакетов), где фразы короче L_{max} должны быть дополнены нулями после индекса токена «>», обозначающего конец слова. Если просто преобразовать фразы

в матрицы путем преобразования слов в их индексы и сделать нулевое заполнение, то тензор будет иметь форму (b_s, W_{max}), где W_{max} – максимальная длина слова, и при индексировании первого измерения будет возвращаться полная последовательность по всем временным шагам. Необходимо иметь возможность индексировать пакет по времени и по всем последовательностям в пакете. Поэтому выполняется операция трансформации входного пакета в размерность (W_{max}, b_s), чтобы индексирование по первому измерению возвращало шаг по времени для всех слов в пакете (рис. 5). Для этого используется транспонирование матрицы E: $F = E^T$.

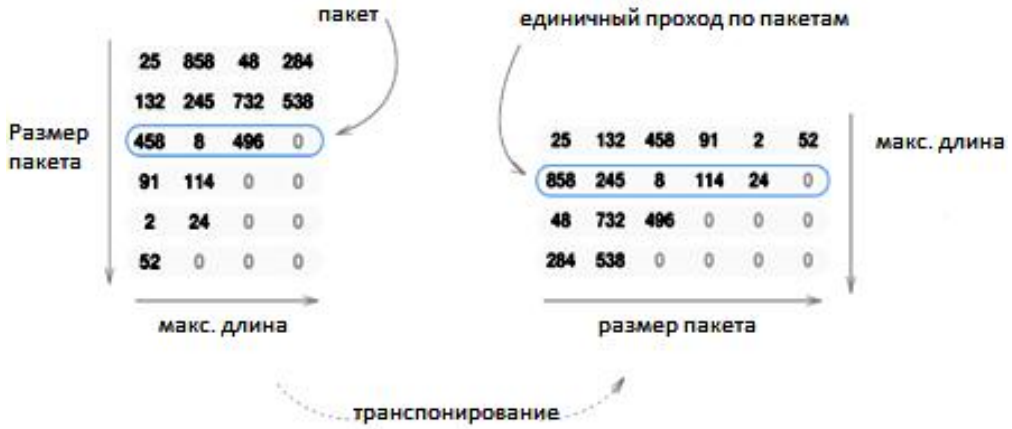

Рисунок 5 – Изображение операции преобразования матрицы индексов слов для мини-пакетного обучения

Алгоритм кодирования состоит в следующем.

1) На вход подаётся $W=\{w_i\}$ (массив слов).

2) Каждое w_i находим в созданном словаре и извлекаем индекс символа. В итоге получается массив индексов символов $U=\{u_m\}$.

3) Информация записывается в матрицу индексов M, длиной $N \times R$, где N – общее количество слов; R – максимальное количество слов в предложении. Матрица M, затем транспонируется и получаем F_{inp}. Помимо матрицы M, формируется матрица L, размером $N \times 1$, содержащая информацию о количестве символов.

4) Аналогичным способом формируем матрицу для транскрипций F_{out}, используя массив транскрипций (T) и максимальную длину транскрипции.

В качестве основной архитектуры для обучения использовалась архитектура Transformer. При обучении нейросети использовались следующие гиперпараметры:

- количество скрытых слоёв: 512;
- размер входных векторов для енкодера: 26;
- размер входных векторов для декодера: 31;
- размер батча: 128;
- количество блоков в енкодере (енкодер переводит входной сигнал в более компактное представление, при этом сохраняя семантическую информацию): 5,
- количество блоков в декодере (восстанавливает исходный сигнал из компактного представления): 3;
- количество заголовков обучающегося внимания: 4;
- функция активации для скрытых слоёв: rectified linear unit;
- функция активации выходного слоя: softmax;
- функция потерь: разреженная кросс-энтропия;
- коэффициент dropout-регуляризации: 0.2;

- функция регуляризации: L2-регуляризация;
- оптимизатор для градиентного спуска: AdamBound;
- коэффициент скорости обучения: 0.0001;
- количество эпох: 100 тыс.

Дополнительно нейросеть была улучшена при помощи следующих модификаций.

1. Техника «принуждения учителя» (teacher forcing) [26]. Это означает, что с некоторой вероятностью, установленной отношением принуждения учителя, мы используем текущее целевое слово в качестве следующего ввода декодера, а не используя текущее предположение декодера. Эта техника действует в качестве обучающих колес для декодера, помогая в более эффективном обучении. Однако принуждение учителя может привести к нестабильности модели во время логического вывода, поскольку у декодера может не быть достаточного шанса по-настоящему создать собственные выходные последовательности во время обучения. Таким образом, мы должны помнить о том, как мы устанавливаем соотношение принуждения учителей, и не обманываться быстрой конвергенцией.

2. Градиентное отсечение (clip gradient) [26]. Это общепринятый метод, направленный на решение проблемы «взрывных градиентов» (vanishing gradients). По сути, обрезая градиенты или устанавливая пороговые значения до максимального значения, мы предотвращаем экспоненциальный рост градиентов и переполнение (равенство градиентов нулю), или превышение крутых обрывов в функции оценивания (рис. 6).

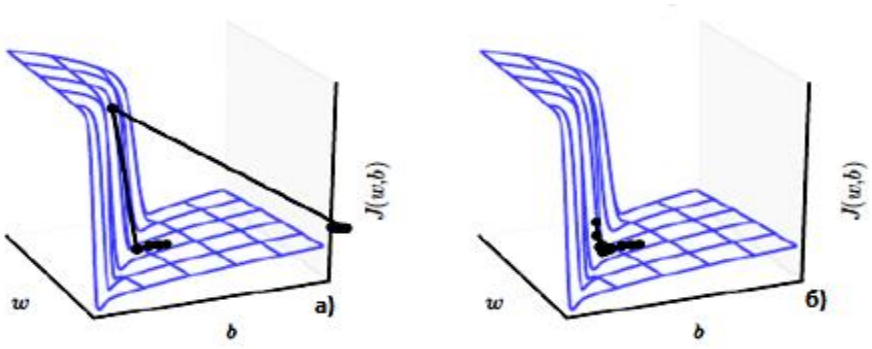

Рисунок 6 – Изображение сравнения функции потерь без градиентного отсечения (а), и с градиентным отсечением (б)

Другой особенностью данной архитектуры является совместное применение обучения с учителем и обучения с подкреплением, реализованным в RL-block. Данная архитектура приведена на рис. 7, где forward-transformer – нейросетевая модель для генерации транскрипции для слов, обученная на парах *x*-*y* (слово-транскрипция); backward-transformer – нейросетевая модель для генерации слов для транскрипций, обученная на парах *y*-*x*; RL-block – механизм обучения с подкреплением; forward_RL-transformer – итоговая нейросетевая модель генерации транскрипции.

Механизм RL-block используется для переопределения вероятностей, т.е. для увеличения правдоподобия «хороших» сценариев (обладающих высокой наградой, reward, R_τ) и понизить правдоподобие «плохих» сценариев (policy gradient):

$$\nabla_\theta J(\theta) = E_{T\sim\rho_\theta(\tau)}\left[\nabla_\theta \log \rho_\theta(\tau) R_\tau\right], \tag{7}$$

где $\rho_\theta(\tau)$ – это вероятность того, что будет реализован сценарий τ при условии параметров модели θ, т.е. функция правдоподобия.

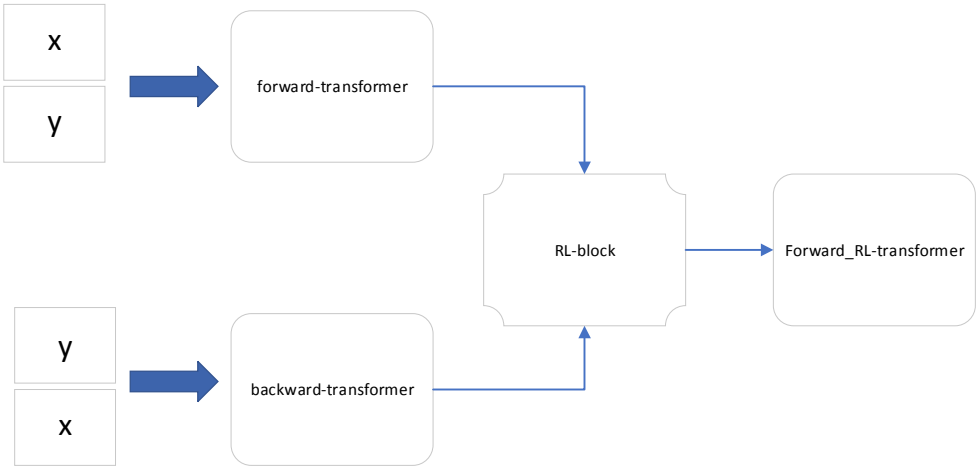

Рисунок 7 – Общая схема обучения модели

Двигаясь вверх по этому градиенту, мы повышаем логарифм функции правдоподобия для сценариев, имеющих большой положительный R_τ.

Данный механизм RL-block заключается в следующем.

1. Дополнительно к forward-transformer обучается backward-transfomer, используя реверсный набор данных для обучения.
2. Инициализируется процесс обучения новой модели (forward_RL-transformer)
3. Используя закодированный набор пар слов и транскрипций к ним, при помощи forward-transformer генерируется набор транскрипций.
4. Вычисляется loss для forward-transformer.
5. Сравнение векторных расстояний. Вычисляется косинусное расстояние между векторами признаков, извлечённых из выходного слоя ($vect_o$) и предпоследнего скрытого слоя forward-transformer ($vect_h$). Вектора признаков сжимаются до минимального размера вектора из двух вышеуказанных векторов:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}, \quad (8)$$

где $A - vect_o$; а $B - vect_h$.

На основе этого вычисляется промежуточный reward (rew_1)

$$rew_1 = \begin{cases} -similarity, & \text{если } similarity < 0 \\ -\log(similarity), & \text{если } similarity > 0 \end{cases} \quad (9)$$

6. Проверка семантической когерентности. На этом этапе промежуточный rewad (rew_2) вычисляется с использованием backward-transformer. Предсказывается слово для транскрипции с соответствующей величиной loss. А также используются данные из forward-transformer:

$$rew_2 = \frac{forw_{loss}}{forw_{res}} + \frac{back_{loss}}{back_{res}}, \quad (10)$$

где $forw_{loss}$, $back_{loss}$ – величина loss при использовании forward-transformer и backward-transformer; $forw_{res}$, $back_{res}$ – результирующий вектор для forward-transformer и backward-transformer.

7. Подсчёт финального reward (rew_{end}):

$$rew_{end} = \frac{rew_1 + rew_2}{2}. \quad (11)$$

8. Формирование списка N размера финальных rewards (rew_{list})

$$rew_{end} = [rew_{end}[0]...rew_{end}[N]]. \quad (12)$$

9. Получение среднего reward (rew_{mean})

$$rew_{mean} = \frac{\sum_{i=1}^{N} rew_{end}[i]}{N}. \quad (13)$$

10. Пересчитывается loss для forward-transformer, на основе которых перестраивается модель.

$$loss_{rl} = forw_{loss} \cdot rew_{mean} \quad (14)$$

11. Используя $loss_{rl}$ для алгоритма обратного распространения ошибки, происходит коррекция весов модели forward_RL-transformer.

Результаты численных исследований

Для обучения модели генерации транскрипции для слов-исключений был собран набор данных, состоящий из слов, отличающихся от фонетических норм русского языка. Данный набор был расширен за счёт генерации парадигм для исходных слов при помощи морфоанализатора pymorphy2. Сгенерированные парадигмы были просмотрены авторами и удалены их неверные варианты. Общий объём слов составил около 10 тыс. Для оценки результатов использовались метрики: WER – отношение количества неверно трансформированных слов к общему количеству слов (Word Error Rate) и PER – отношение количества неверно трансформированных символов к общему количеству символов (Phoneme Error Rate).

Результаты тестирования разработанной нейросетевой модели для генерации транскрипций слов-исключений приведены на рис. 8, где рис. 8а – зависимость loss-функций от количества эпох; рис. 8б – зависимость метрик WER от количества эпох; рис. 8в – зависимость метрик PER от количества эпох.

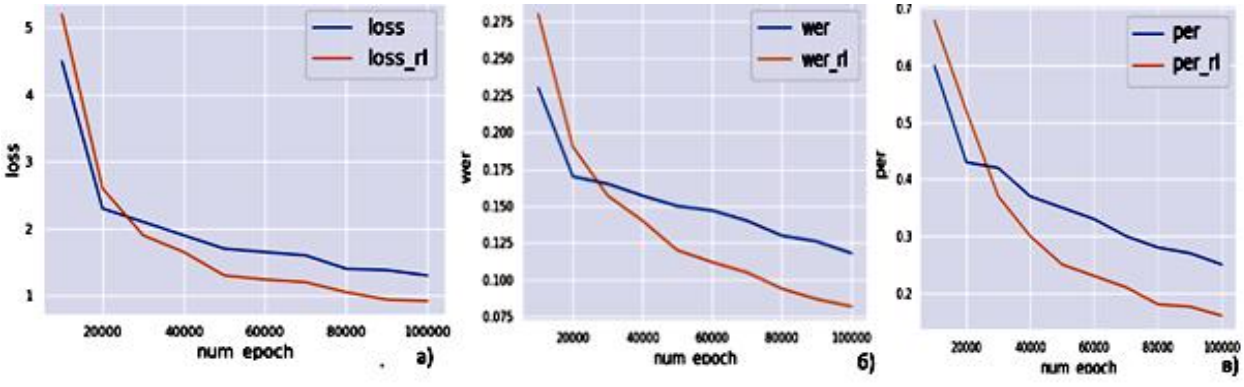

Рисунок 8 – Результаты тестирования сети PhonExcNN

Выводы

Проведено исследование относительно фонемного состава русского языка, в результате которого были выделены особенности его произношения.

Проанализированы существующие подходы к задаче фонемного транскрибирования генерации транскрипции слов, выделены преимущества и недостатки каждого подхода.

Обучена и протестирована работа нейросетевой модели генерации транскрипций для слов-исключений. Архитектура модифицирована за счёт применения техник clip gradient, teacher forcing и увеличения количества блоков в енкодере и декодере. Как показали численные исследования, предложенная техника модернизации моделей типа sequence-to-sequence на основе внесения изменений в структуру алгоритма построения позволила повысить точность обученной модели генерации транскрипций для слов-исключений по критерию PER на 9%, по критерию WER на – 3%.

Список литературы

1. Панов М. В. Современный русский язык. Фонетика: учебник для ун-тов [Текст] / Панов М. В. – М. : Высш. Школа, 1979. – 256 с.
2. Валгина Н. С. Современный русский язык [Текст] / Н. С. Валгина, Д. Э. Розенталь, М. И. Фомина. – Логос, 2006.
3. Малышева Е. Г. Современный русский язык. Фонетика. Орфоэпия: учебное пособие [Текст] / Е. Г. Малышева, О. С. Рогалева. – Омск : Изд-во Ом. гос. ун-та, 2012.
4. Князев С. В. Современный русский литературный язык: фонетика, орфоэпия, графика, орфография [Текст] / С. В.Князев, С. К. Пожарицкая. – 2011. – С. 432–432.
5. Гируцкий А. А. Введение в языкознание [Текст] / А. А. Гируцкий ; [рецензенты: к.фил.н., доц. Е. С. Садовская, к.фил.н., доц. Ж. С. Спливеня]– Минск : Вышэйшая школа, 2016. – 238 с.
6. Грищенко А. Фонетика современного русского литературного языка (Фонетика. Фонология. Орфоэпия. Графика. Орфография) [Текст] / А. Грищенко, М. Попова. – Litres, 2019.
7. Кривнова О.Ф. Многофункциональный автоматический транскриптор русских текстов [Текст] / О. Ф. Кривнова, Л.М. Захаров, Г.С. Строкин // Труды Международного конгресса исследователей русского языка. – М. – 2001.
8. Hunnicutt S. Grapheme-to-phonemerules: Areview [Текст] / S. Hunnicutt // Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, Sweden, QPSR 2-3. – 1980. – Pp. 38–60.
9. Смирнов В.А. Функция лингвистического процессора в системе автоматического анализа неструктурированной речевой информации [Текст] / В.А. Смирнов, М.Н. Гусев, М.П. Фархадов// Автоматизация и современные технологии. – № 8. – 2013. – С. 20–28.
10. Bisani M. Joint-sequence models for grapheme-to-phoneme conversion [Текст] / M. Bisani, H. Ney // SPECOM. – 2008.
11. Novak J. WFST-based Grapheme-to-Phoneme Conversion: Open Sourse Tools for Alignment, Model-Building and Decoding [Текст] /J. Novak, N. Minematsu, K. Hirose // Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing. – 2012. – Pp.45–49.
12. Sun, Hao, et al. Token-Level Ensemble Distillation for Grapheme-to-Phoneme Conversion[Текст] /Sun, Hao, et al. //arXiv preprint arXiv:1904.03446. – 2019.
13. Yolchuyeva, Sevinj, GézaNémeth, and BálintGyires-Tóth. Transformer based Grapheme-to-Phoneme Conversion[Текст] /Yolchuyeva, Sevinj, GézaNémeth, and BálintGyires-Tóth // Proc. Interspeech 2019. – 2019. – 2095-2099.
14. Karanikolas, Nikitas N. Machine learning of phonetic transcription rules for Greek [Текст] / Karanikolas, Nikitas N. // AIP Conference Proceedings. – Vol. 2116, No. 1. – AIP Publishing, 2019.
15. Huang, Qiang. Detecting Mismatch Between Speech and Transcription Using Cross-Modal Attention}}[Текст] / Huang, Qiang, and Thomas Hain // Proc. Interspeech 2019. – 2019. – Pp. 584-588.
16. Jůzová, Markéta. Using Auto-Encoder BiLSTM Neural Network for Czech Grapheme-to-Phoneme Conversion [Текст] / Jůzová, Markéta, and Jakub Vít // International Conference on Text, Speech, and Dialogue. – Springer, Cham, 2019.

17. Ponomareva, Maria, et al. Automated word stress detection in russian [Текст] / Ponomareva, Maria, et al. // arXiv preprint arXiv:1907.05757. – 2019.
18. Algorithms for automatic accentuation and transcription of russian texts in speech recognition systems [Текст] / Yakovenko O., Bondarenko I., Borovikova M., Vodolazsky D. // Karpov A., Jokisch O., Potapova R. (eds.) SPECOM 2018. LNCS (LNAI). – Vol. 11096. – Pp. 768–777.
19. Кипяткова И.С. Модуль фонематического транскрибирования для системы распознавания разговорной русской речи [Электронный ресурс] / И.С. Кипяткова, А.А. Карпов // Искусственный интеллект. – 2008. – URL: http://www.nbuv.gov.ua/ portal/natural/ii/2008_4 /JournalAI_2008_4/ Razdel9/00_Kipyatkova_Karpova.pdf
20. Yanushevskaya Irena; Bunčić, Daniel. Russian [Текст] / Yanushevskaya, Irena; Bunčić, Daniel //Journal of the International Phonetic Association. –2015. – № 45 (2). – Pp. 221–228, doi:10.1017/S0025100314000395.
21. Sutskever, I., O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks [Текст] / Sutskever, I., O. Vinyals, and Q. V. Le. – Advances in NIPS, 2014.
22. Toshniwal, Shubham, and Karen Livescu. Jointly learning to align and convert graphemes to phonemes with neural attention models. 2016 [Текст] / Toshniwal, Shubham, and Karen Livescu // IEEE Spoken Language Technology Workshop (SLT). – IEEE, 2016.
23. G2PwithTensorflow [Электронный ресурс]. – URL: https://github.com/cmusphinx/g2p-seq2seq (дата обращения: 16.05.2019)
24. Lee, Younggun, and Taesu Kim. Learning pronunciation from a foreign language in speech synthesis networks." [Текст]/Lee, Younggun, and Taesu Kim //arXiv preprint arXiv:1811.09364, – 2018.
25. Attention is all you need. [Текст] / Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ...&Polosukhin, I. // Advances in neural information processing systems. – 2017.Pp. 5998-6008.
26. Morphological analyzer and generator for Russian and Ukrainian languages [Текст] /Korobov M. // International Conference on Analysis of Images, Social Networks and Texts. – Springer, Cham, 2015. – С. 320-332.

References

1. Panov M. V. *Sovremennyirusskyiyazyk. Fonetika :Uchebnikdlya un-tov* [Century. Modern Russian language. Phonetics: A Textbook for University], M.,Vysh. Shkola, 1979, pp. 256.
2. Valgina N. S., Ditmar E. R., Fomina M. I. *Sovremennyi russkyi* yazyk [Modern Russian language], Logos, 2006.
3. Malysheva, E. G., Rogaleva O. S. *Sovremennyi russkyi yazyk. Fonetika. Orfoepiya: yuchebnoe posobie* [Modern Russian language. Phonetics. Orthoepy: study guide], Omsk, Izd-vo Om.gos. un-ta, 2012.
4. Knyazev S. V., Pozharitskaya S. K. *Sovremennyi russkyi literaturnyi yazyk: fonetika, orfoepiya, grafika, orfographiya*[Modern Russian literary language: phonetics, spelling, graphics, spelling], 2011, 432-432.
5. Girutskiy A. A. *Vvedenie v yazykoznanie* [Introduction to linguistics] / A. A. Girutskiy [rescensenty: k.fil.n., dots. E. S. Sadovskaya, k.fil.n., dots. Zh. S. Splivenya], Minsk, Vysheishayashkola, 2016, 238 s.
6. Gritschenko A., Popova M. *Fonetika sovremennogo russkogo literaturnogo yazyka (Fonetika. Fonologiya. Orfoepiya. Grafika. Orfografiya)* [Phonetics of the modern Russian literary language (Phonetics. Phonology. Orthoepy. Graphics. Spelling)],Litres, 2019.
7. Krivnova O.F., Zaharov L.M., Strokin G.S. Mnogofunkcionalnyj avtomaticheskij transkriptor russkih tekstov [Multifunctional automatic transcript of Russian texts].*Trudy Mezhdunarodnogo kongressa issledovatele jrusskogo yazyka* [Proceedings of the International Congress of Russian Language Researchers], M., 2001.
8. Hunnicutt S. Grapheme-to-phoneme rules: A review. *Speech Transmission Laboratory, Royal Institute of Technology*, Stockholm, Sweden, QPSR 2-3, 1980, pp. 38-60.
9. Smirnov V.A., Gusev M.N., Farhadov M.P. Funkciya lingvisticheskogo processora v sisteme avtomaticheskogo analiza nestrukturirovannoj rechevoj informacii [Function of the linguistic processor in the system of automatic analysis of unstructured speech information]. *Avtomatizaciya i sovremennye tehnologii* [Automation and modern technology], No. 8, 2013, pp. 20-28.
10. Bisani M., Ney H. Joint-sequence models for grapheme-to-phoneme conversion // SPECOM. – 2008.
11. Novak J., Minematsu N., Hirose K. WFST-based Grapheme-to-Phoneme Conversion: Open Sourse Tools for Alignment, Model-Building and Decoding.*Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, 2012, pp.45-49.
12. Sun, Hao, et al. Token-Level Ensemble Distillation for Grapheme-to-Phoneme Conversion.*arXiv preprint arXiv:*1904.03446 (2019).

13. Yolchuyeva, Sevinj, GézaNémeth, and BálintGyires-Tóth. Transformer based Grapheme-to-Phoneme Conversion. *Proc. Interspeech* 2019 (2019): 2095-2099.
14. Karanikolas, Nikitas N. Machine learning of phonetic transcription rules for Greek.*AIP Conference Proceedings*. Vol. 2116. No. 1. AIP Publishing, 2019.
15. Huang, Qiang, and Thomas Hain. Detecting Mismatch Between Speech and Transcription Using Cross-Modal Attention}}.*Proc. Interspeech* 2019 (2019): 584-588.
16. Jůzová, Markéta, and JakubVít. Using Auto-Encoder BiLSTM Neural Network for Czech Grapheme-to-Phoneme Conversion.*International Conference on Text, Speech, and Dialogue*. Springer, Cham, 2019.
17. Ponomareva, Maria, et al. Automated word stress detection in russian.*arXiv preprint arXiv*:1907.05757 (2019)
18. Yakovenko, O., Bondarenko, I., Borovikova, M., Vodolazsky, D.: Algorithms for automatic accentuation and transcription of russian texts in speech recognition systems. In: Karpov, A., Jokisch, O., Potapova, R. (eds.) *SPECOM 2018. LNCS (LNAI)*, vol. 11096, pp. 768–777.
19. Kipyatkova I.S., Karpov A.A. Modul fonematicheskogo transkribirovaniya dlya sistemy raspoznavaniya razgovornoj russkoj rechi [Phonemic transcription module for the recognition system of spoken Russian speech].*Iskusstvennyjintellect* [Artificial Intelligence], 2008. URL:http://www.nbuv.gov.ua/portal/natural/ii/2008_4/JournalAI_2008_4/Razdel9/00_Kipyatkova_Karpova.pdf
20. Yanushevskaya, Irena; Bunčić, Daniel (2015), Russian (PDF), *Journal of the International Phonetic Association*, 45 (2): 221–228, doi:10.1017/S0025100314000395
21. Sutskever, I., O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *Advances in NIPS* (2014).
22. Toshniwal, Shubham, and Karen Livescu. Jointly learning to align and convert graphemes to phonemes with neural attention models. 2016 *IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2016.
23. *G2P with Tensorflow*.URL: https://github.com/cmusphinx/g2p-seq2seq (дата обращения: 16.05.2019)
24. Lee, Younggun, and Taesu Kim. Learning pronunciation from a foreign language in speech synthesis networks.*arXiv preprint arXiv*:1811.09364 (2018).
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ...&Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems* (pp. 5998-6008).
26. Korobov M. Morphological analyzer and generator for Russian and Ukrainian languages. *International Conference on Analysis of Images, Social Networks and Texts*, Springer, Cham, 2015, pp. 320-332.

RESUME

Ya. S. Pikalyov, T. V. Yermolenko

System of automatic transcription generation of Russian-language words exceptions on the basis of deep learning

Since the recognized units in speech recognition systems are phonetic level units, it is necessary to create a dictionary containing words with their orthographic and phonemic representation. Such a dictionary is developed using canonical phonetic transcription rules. Word transcription generation is one of the most important steps that affect the effectiveness of speech recognition. Russian phonetics and orthoepy rules do not apply to the algorithm for obtaining transcription for exception words. Therefore, it is an urgent task to build models for automatic transcription generation for exception words that take into account the features of Russian phonetics.

The article considers approaches to automatic transcription generation, their advantages and disadvantages. Among the algorithms for generating transcriptions of words belonging to the statistical modeling group, the neural network approach shows the best results if there is a large amount of training data.

The training data is usually a dictionary of words with their phonemic transcriptions.

The Transformer neural network model based on the seq2seq model, which uses separate models of the encoder, converting words of the input sentence into one or more vectors in a certain space, and the decoder, generating a sequence of words from these vectors, is well established in natural language processing tasks. The encoder in the Transformer uses layers with a masking multilayer attention mechanism.

In the paper a model based on the Transformer architecture with some modifications is proposed for automatic generation of transcription of exception words. The model used the “teacher forcing” technique to train the decoder, which is to use the current target word as the decoder’s next input, rather than the decoder’s current assumption. This made it possible to increase the efficiency of decoder training. In addition, gradient clipping was used to solve the problem of “explosive gradients”. Another feature of the proposed architecture is the combined use of teacher training and reinforcement learning, which is used to redefine probabilities on the output layer. This technique increases the likelihood of “good” scenarios (which have a high reward) and lowers the likelihood of “bad” scenarios, which increases the accuracy of the model.

To train the neural network, a word encoding algorithm was developed to optimize the learning process of the mini-batch type of training, and a dictionary containing more than 5 million word forms was created, as well as phonetic dictionaries of exception words available on the Network, which were supplemented with paradigms.

The results of numerical studies show that the proposed technique of modernization models such as sequence-to-sequence based changes in the structure of the algorithm allowed to increase the accuracy of the trained model generating transcriptions for exception words for PER-exclusion words by 9%, and by 3% according to WER criterion.

РЕЗЮМЕ

Я. С. Пикалёв, Т. В. Ермоленко

Система автоматической генерации транскрипций русскоязычных слов-исключений на основе глубокого обучения

Поскольку в системах распознавания слитной речи распознаваемыми единицами являются единицы фонетического уровня, то возникает необходимость создания словаря, содержащего слова с их орфографическим и фонематическим представлением. Такой словарь разрабатывается с использованием канонических фонетических правил транскрибирования. Генерация транскрипции слов является одним из важных этапов, влияющих на эффективность распознавания речи. Алгоритм получения транскрипции для слов-исключений не подчиняется правилам фонетики и орфоэпии русского языка, поэтому построение моделей автоматической генерации транскрипций для слов-исключений, учитывающих особенности фонетики русского языка, является актуальной задачей.

В статье рассмотрены подходы к автоматической генерации транскрипций, их преимущества и недостатки. Среди алгоритмов генерации транскрипций слов, относящихся к группе статистического моделирования, наилучшие результаты показывает нейросетевой подход при наличии обучающих данных большого объема. В качестве обучающих данных обычно выступает словарь слов с их фонемными транскрипциями.

В задачах обработки естественного языка хорошо зарекомендовала себя нейросетевая модель Transformer, основанная на seq2seq модели, которая использует отдельные модели енкодера, преобразовывая слова входного предложения в один или больше векторов в определенном пространстве, и декодера, генерируя из этих векторов последовательность слов. Енкодер в Transformer использует слои с механизмом маскирующего многослойного внимания.

В работе для автоматической генерации транскрипции слов-исключений предлагается модель на основе архитектуры Transformer с некоторыми модификациями.

В модели для обучения декодера использовалась техника «teacher forcing», которая заключается в использовании текущего целевого слова в качестве следующего ввода декодера, а не текущего предположения декодера. Это позволило повысить эффективность обучения декодера. Кроме того, использовалось градиентное отсечение для решения проблемы «взрывных градиентов». Еще одной особенностью предложенной архитектуры является совместное применение обучения с учителем и обучения с подкреплением, которое используется для переопределения вероятностей на выходном слое. Эта техника увеличивает правдоподобие «хороших» сценариев (обладающих высокой наградой) и понижает правдоподобие «плохих» сценариев, что повышает точность модели.

Для обучения нейросети был разработан алгоритм кодирования слов, направленный на оптимизацию процесса обучения модели при помощи мини-пакетного типа обучения, а также создан словарь, содержащий более 5 млн уникальных словоформ, доступные в Сети фонетические словари слов-исключений, которые были дополнены парадигмами.

Результаты численных исследований показали, что предложенная техника модернизации моделей типа sequence-to-sequence на основе внесения изменений в структуру алгоритма построения позволила повысить точность обученной модели генерации транскрипций для слов-исключений по критерию PER на 9%, по критерию WER на – 3%.

Статья поступила в редакцию 13.11.2019.