УДК 004.522 **DOI** 10.15622/sp.58.4

Н.М. Марковников, И.С. Кипяткова

АНАЛИТИЧЕСКИЙ ОБЗОР ИНТЕГРАЛЬНЫХ СИСТЕМ РАСПОЗНАВАНИЯ РЕЧИ

Марковников Н.М., Кипяткова И.С. **Аналитический обзор интегральных систем распознавания речи.**

Аннотация. Приведен аналитический обзор разновидностей интегральных (end-to-end) систем для распознавания речи, методов их построения, обучения и оптимизации. Рассмотрены варианты моделей на основе коннекционной временной классификации (CTC) в качестве функции потерь для нейронной сети, модели на основе механизма внимания и шифратор-дешифратор моделей. Также рассмотрены нейронные сети, построенные с использованием условных случайных полей (CRF), которые являются обобщением скрытых марковских моделей, что позволяет исправить многие недостатки стандартных гибридных систем распознавания речи, например, предположение о том, что элементы входных последовательностей звуков речи являются независимыми случайными величинами. Также описаны возможности интеграции с языковыми моделями на этапе декодирования, демонстрирующие существенное сокращение ошибки распознавания для интеграционных моделей. Описаны различные модификации и улучшения стандартных интегральных архитектур систем распознавания речи, как, например, обобщение коннекционной классификации и использовании регуляризации в моделях, основанных на механизмах внимания. Обзор исследований, проводимых в данной предметной области, показывает, что интегральные системы распознавания речи позволяют достичь результатов, сравнимых с результатами стандартных систем, использующих скрытые марковские модели, но с применением более простой конфигурации и быстрой работой системы распознавания как при обучении, так и при декодировании. Рассмотрены наиболее популярные и развивающиеся библиотеки и инструментарии для построения интегральных систем распознавания речи, такие как TensorFlow, Eesen, Kaldi и другие. Проведено сравнение описанных инструментариев по критериям простоты и доступности их использования для реализации интегральных систем распознавания речи.

Ключевые слова: автоматическое распознавание речи, интегральные системы, нейронные сети, глубокое обучение.

1. Введение. В настоящее время с увеличением вычислительной мощности компьютеров задача распознавания речи становится все более востребованной. Распознавание речи используется в таких областях, как: управление интерфейсом множества приложений (навигаторы, мессенджеры и т.д.), распознавание телефонных разговоров, генерация речи и так далее. Существуют качественные стандартные модели распознавания речи, показывающие хорошие результаты и состоящие из множества различных частей. Но в них все компоненты обучаются независимо, и ошибки в одних компонентах могут вызывать ошибки в других. Сценарий стандартной системы состоит из множества шагов, что требует гигабайты памяти для хранения, например, обученных языковых моделей, и не позволяет использовать системы локально на различных устройствах, а появляется необходимость удаленных вычислений на серверах. Кроме того,

использование широко применяемых для акустического моделирования скрытых марковских моделей (СММ) имеет недостатки: используется предположение о том, что наблюдения являются независимыми случайными величинами и применяются «слабые модели» — модели Маркова первого порядка.

В последнее время получил распространение другой подход: обучение выполняется так, что только одна модель может выдавать нужный выход без использования других компонент. Такие модели называются *интегральными* (*end-to-end*). Обычно в качестве интегральных моделей служат глубокие искусственные нейронные сети (ИНС). Отметим преимущества такого подхода перед стандартным:

– интегральные модели проще реализовать, так как они могут включать в себя только одну нейронную сеть, которая может быть написана только с использованием одного фреймворка и обучена только с помощью градиентного спуска и одной функции потерь; это уменьшает вероятность появления ошибок в коде программы;

– интегральные модели демонстрируют лучшую производительность (скорость, а иногда и точность);

– интегральные модели потенциально требуют меньший объем памяти компьютера, что позволяет использовать их на мобильных устройствах локально.

Недостатком таких моделей является потребность в большом количестве размеченных данных для обучения, что на некоторых типах задач может быть проблематично.

В данном обзоре рассматриваются три вида интегральных моделей на основе глубоких ИНС для распознавания речи: на основе коннекционной временной классификации, шифратор-дешифратор модели с использованием механизма внимания и модели, использующие условные случайные поля. Но сначала рассмотрим архитектуру стандартных систем.

2. Стандартная система распознавания речи. Цель автоматического распознавания речи — преобразование звукового сигнала S в последовательность слов W. Эту задачу можно сформулировать [1] как поиск наиболее вероятной последовательности слов по входному сигналу S:

$$W^* = \arg\max_{W \in \Omega} P(W \mid S), \tag{1}$$

где Ω — множество гипотез.

Обычно система распознавания речи разбивает задачу на три шага, как показано на рисунке 1: выделение признаков, акустическое

моделирование и декодирование последовательности. Рассмотрим каждый шаг более подробно.

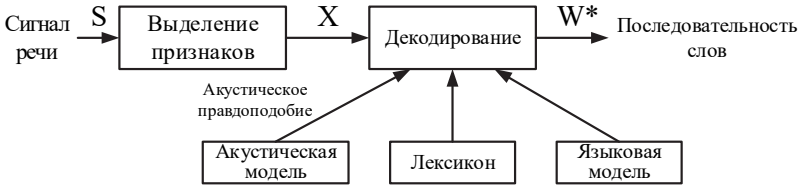

Рис. 1. Стандартная система распознавания речи

2.1. Выделение признаков. На данном шаге осуществляется выделение признаков X из сигнала речи S в зависимости от задачи и речевых особенностей и уменьшение пространства векторов признаков. Два наиболее популярных типа признаков: мел-частотные кепстральные коэффициенты (MFCC) [2] и коэффициенты перцептивного линейного предсказания (perceptual linear prediction cepstral coefficient; PLP) [3]. Получение данных признаков состоит из следующих этапов:

1. преобразование сегмента сигнала речи в множество частот (например, с помощью дискретного преобразования Фурье);
2. применение различных фильтров;
3. применение нелинейной функции ($\ln(\cdot)$ или $\sqrt[3]{\cdot}$);
4. применение различных преобразований для уменьшения размерности некореллированных признаков (дискретное косинусное преобразование или авторегрессионные модели).

Описанный процесс моделирует только локальное изменение сигнала в окне длительностью, как правило, 20-30 мс. Но сигнал распространен во времени, поэтому для моделирования временных изменений сигнала используют первую и вторую производные признаков.

2.2. Акустическое моделирование. Акустическое моделирование используется для построения статистических зависимостей между признаками и лингвистическими единицами, например, фонемами.

2.3. Декодирование последовательности. Декодирование последовательности преобразует последовательность признаков X в последовательность слов W. Этот шаг можно описать следующим образом:

$$W^* = \arg\max_{W\in\Omega} P(W \mid X) = \arg\max_{W\in\Omega} \frac{P(X \mid W)P(W)}{P(X)} = \\ = \arg\max_{W\in\Omega} P_A(X \mid W)P_L(W), \tag{2}$$

где $P_L(W)$ — априорная вероятность, получаемая с помощью языковых моделей (ЯМ), а $P_A(X \mid W)$ — функция правдоподобия на основе акустических моделей (АМ).

2.4. Системы, основанные на скрытых марковских моделях. Популярным подходом к построению систем распознавания речи является использование скрытых марковских моделей (СММ) [4]. В таких системах функцию правдоподобия на основе акустических моделей, используя теорему Байеса и алгоритм Витерби, можно описать следующим образом:

$$P_A(X \mid W) = \max_{Q \in \Xi} \prod_{t=1}^{T} P_e(x_t \mid q_t = i) P_{tr}(W \mid q_{t-1} = j), \tag{3}$$

где $X = \{x_1, \ldots, x_t, \ldots, x_T\}$ — последовательность признаков, $Q = \{q_1, \ldots, q_t, \ldots, q_T\} \in \Xi$ — множество скрытых состояний СММ, каждое из которых описывается вероятностью наблюдений из распределения вероятностей $P(x_t \mid q_t)$, где состояния соответствуют классам $i \in \{1, \ldots, I\}$. Также предполагается, что x_t зависят только от текущего состояния q_t, и q_t зависит только от предыдущего состояния q_{t-1}. $P_e(x_t \mid q_t = i)$ — вероятности наблюдений для класса i, $P_{tr}(q_t = i \mid q_{t-1} = j)$ — вероятности переходов между классами i и j момент времени t.

Существуют два основных подхода к определению вероятностей наблюдений: смеси гауссовских распределений плотностей вероятностей (Gaussian Mixture Model; GMM) и искусственные нейронные сети (ИНС; Artificial Neural Networks; ANN).

В системе, использующей смеси гауссовских распределений плотностей вероятностей, вероятности наблюдений определяются как:

$$P_e(x_t \mid q_t = i) = \sum_{j=1}^{J} c_{ij} \mathrm{N}(x_t, \mu_{ij}, \Sigma_{ij}), \tag{4}$$

где J — число компонент смеси, c_{ij} — вес гауссовского распределения $\mathrm{N}(x_t, \mu_{ij}, \Sigma_{ij})$, μ_{ij} и Σ_{ij} — элемент вектора математических ожиданий и ковариационная матрица соответственно.

В гибридной СММ/ИНС модели вероятности наблюдений вычисляются с помощью нейронной сети. ИНС вычисляет вероятности в зависимости от класса $P(x_t \mid q_t = i)$. Так, пользуясь теоремой Байеса, можно вычислить вероятности наблюдений:

$$P_e(x_t \mid q_t = i) = \frac{P(x_t \mid q_t = i)}{P(x_t)} = \frac{P(q_t = i \mid x_t)}{P(q_t = i)}. \tag{5}$$

Схема гибридной СММ/ИНС модели изображена на рисунке 2. Используются различные архитектуры ИНС для построения гибридных моделей, например:

– многослойные перцептроны (Multilayer Perceptron; MLP) или глубокие нейронные сети (Deep Neural Networks; DNN) [5, 6];

– свёрточные нейронные сети (Convolutional Neural Networks; CNN) [7-9];

– рекуррентные нейронные сети (Recurrent Neural Networks; RNN) [10];

– нейронные сети с длительной кратковременной памятью (Long Short Term Memory; LSTM) [11];

– управляемый рекуррентный блок (Gated Recurrent Unit; GRU) [12];

– двунаправленные рекуррентные нейронные сети (Bidirectional Recurrent Neural Networks BRNN) [13];

– остаточные сети (Residual Networks) [14, 15].

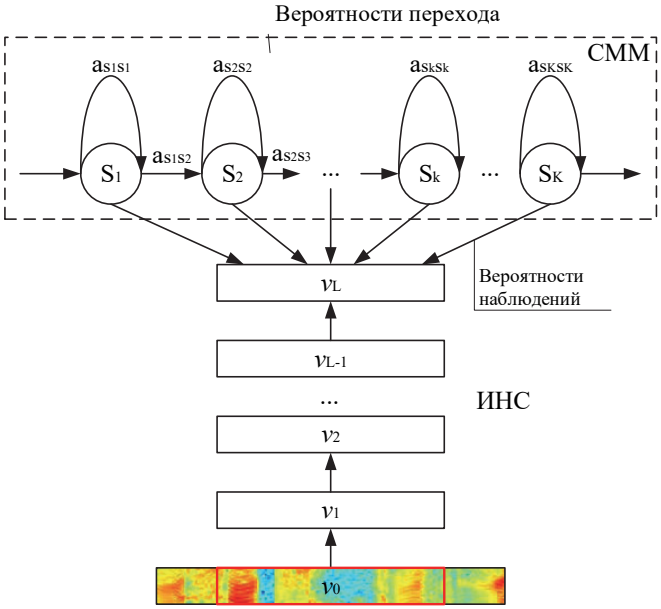

Рис. 2. Гибридная СММ/ИНС модель

Подробный обзор ИНС, используемых для построения гибридных систем распознавания речи, представлен в [16]. Также применяются различные типы инициализации матрицы весов (например, глу-

бокими сетями доверия (Deep Belief Networks; DBN) [17]), регуляризации (например, дропаут (dropout) [18]), нормализации (батч-нормализация (batch-normalization) [19]) и так далее.

2.5. Метрики. Для измерения качества работы системы распознавания слитной речи обычно используют количество неверно распознанных слов (Word Error Rate; WER), количество неверно распознанных фонем (Phoneme Error Rate; PER) или количество неверно распознанных символов (Character Error Rate; CER). Эти метрики вычисляются с помощью расстояния Левенштейна [20] между данной и полученной последовательности слов, фонем или символов:

$$WER / PER / CER = \frac{D + S + I}{N} \cdot 100\%, \quad (6)$$

где N — общее число слов, фонем или символов в данной последовательности, D — число удалений, S — число замен, I — число вставок.

3. Интегральные системы распознавания речи. Во многих работах было показано, что использование нейронных сетей на каждом шаге сценария стандартной системы распознавания речи улучшает качество ее работы. Так, например, в [21] языковые модели были обучены с помощью RNN, в [22] словарь был получен с помощью LSTM сетей, в [4] глубокие нейронные сети показали высокие результаты для построения акустических моделей, в [23] был представлен метод выделения признаков с помощью ограниченных машин Больцмана [24]. Следовательно, появилась идея использовать ИНС на всех этапах распознавания речи.

3.1. Описание подхода. Как уже было сказано, многие системы содержат множество компонент, обучаемых независимо друг от друга, которые затем объединяются в цепочку для получения нужного результата. Например, чтобы обучить некоторого робота двигаться в нужном направлении на основе визуальных признаков, первый компонент может быть обучен преобразовывать визуальные данные в некоторое промежуточное представление, которое будет принимать другой компонент и выдавать команды для робота. Виды таких пошаговых сценариев нужны тогда, когда вход и выход модели имеют разную «природу», например:

– вход — звуковой сигнал, выход — текст;

– вход — значения пикселей изображения, выход — текстовое описание изображения;

– вход — значения пикселей изображения, выход — команды для робота [25] и так далее.

Подход, когда обучение выполняется так, что только одна модель может выдавать нужный выходной результат без использования

других компонент, называется интегральным. А модель, реализующую этот подход — интегральной моделью.

3.2. Интегральный подход в распознавании речи. В случае распознавания речи интегральный подход пытается вычислить $P(W \mid X)$ «глобально». Пусть вход представляет собой последовательность звуковых признаков $X = (x_1, \ldots, x_T)$, а соответствующая ему последовательность слов — $W = w_m = (w_1, \ldots, w_M)$. Так, нейронная сеть вычисляет вероятности $P(\cdot \mid x_1), \ldots, P(\cdot \mid x_T)$, где аргументами вероятностей являются не сами последовательности слов, а некоторые их представления (далее — метки). На рисунке 3 изображена схема работы интегральной системы. На данный момент существует несколько методов реализации интегральных моделей. Далее рассмотрим три подобных метода:

1. Модели на основе коннекционной временной классификации.
2. Шифратор-дешифратор модели, основанные на механизме внимания.
3. Условные случайные поля.

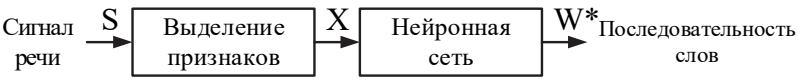

Рис. 1. Интегральная система распознавания речи

3.3. Модели на основе коннекционной временной классификации. Нейронные сети в распознавании речи обычно обучаются с помощью отдельных фрагментов звуковых записей речи. Для этого требуется выделять отдельные метки, соответствующие для каждого кадра, что влечет за собой необходимость выравнивания звуковой дорожки и транскрипции. Однако выравнивание надежно только после обучения нейронной сети, что приводит к циклической зависимости между сегментацией и распознаванием (известной как парадокс Сайре [26] в тесно связанной области распознавания рукописного ввода). Более того, в задачах распознавания речи, основанных только на транскрипции слов, выравнивание не приносит пользы.

Коннекционная временная классификация (Connectionist Temporal Classification; CTC) [27] — это функция, которая позволяет рекуррентным нейронным сетям обучаться для распознавания последовательности слов без начального выравнивания входных и выходных последовательностей.

Этап обучения. Опишем подход, в котором CTC-функция используется в качестве функции потерь для обучения нейронной сети. Выходной слой нейронной сети содержит по одному блоку для каждо-

го символа выходной последовательности (букв, фонем, знаков препинания, нот) и еще один для дополнительного символа «пропуск» («blank»), соответствующего пустому выходному символу. Выходной вектор w_m нормализуется с помощью softmax [28] функции, которая интерпретируется как вероятность появления символа (или «пропуска») с индексом k в момент времени m:

$$P(k,m \mid x) = \frac{\exp(w_m^k)}{\sum\limits_{k'=0}^{|w_m|} w_m^{k'}}, \tag{7}$$

где w_m^k — k-ый элемент w_m, а — длина слова w_m. Пусть, α — последовательность из индексов «пропусков» и символов длины T для выравнивания. Вероятность $P(\alpha \mid x)$ можно представить как произведение вероятностей появления символов в каждый момент времени:

$$P(\alpha \mid x) = \prod_t^T P(\alpha_t, t \mid x). \tag{8}$$

Для данной выходной последовательности $|w_m|$ существует столько возможных выравниваний, сколько способов расставить «пропуски» между символами. Пусть «—» означает «пропуск». Например, выравнивания (а,—,б,в,—,—) и (—, —, а, —, б, в) соответствуют последовательности (а,б,в). Когда одинаковые символы появляются последовательно, то эти повторы удаляются: (а,б,б,б,в,в) и (а,—,б,—,в,в) соответствуют (а,б,в). Обозначим, что B — оператор, который удаляет сначала все повторы, а затем — «пропуски». Так, полная вероятность выходной последовательности w равна сумме вероятностей всех возможных соответствующих выравниваний:

$$P(w \mid x) = \sum_{\alpha \in \mathrm{B}^{-1}(w)} P(\alpha \mid x), \tag{9}$$

где B^{-1} — оператор, обратный к B.

Эта сумма по всем возможным выравниваниям позволяет нейронной сети тренироваться на несегментированных данных. То есть, не зная точное расположение меток, мы суммируем по всем расположениям, где они могут быть. Эта сумма может быть вычислена с помощью динамического программирования [27]. Пусть w^* — целе-

вая последовательность слов, тогда нейронная сеть может быть обучена минимизировать CTC функцию:

$$CTC(x) = -\log P(w^* \mid x). \tag{10}$$

Нейронная сеть может быть обучена с помощью любого оптимизационного алгоритма, использующего градиент. На рисунке 4 представлена схема CTC модели, где шифратор может быть DNN, LSTM, BLSTM, CNN или любой другой разновидностью нейронных сетей. В [27] предложен CTC алгоритм прямого-обратного хода, который использует алгоритм динамического программирования, похожий на алгоритм прямого обратного хода для СММ [29]. Основная идея этого алгоритма в том, что сумма по всем выравниваниям разбивается на сумму по выравниваниям соответствующих префиксам их выходных последовательностей. Эта сумма может быть эффективно вычислена с помощью рекурсивных прямых и обратных переменных.

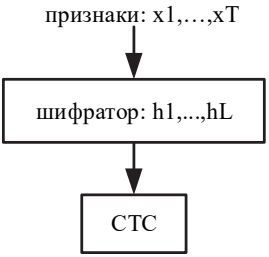

Рис. 4. CTC система распознавания речи

Также в [27] была предложена метрика количества неверных меток (label error rate, LER) для временного классификатора h как среднее нормализованное расстояние Левенштейна между выходом классификатора и истинным результатом:

$$LER(h, S') = \frac{1}{|S'|} \sum_{(x,w) \in S'} \frac{dist(h(x), w)}{|w|}, \tag{11}$$

где $dist(p, q)$ — расстояние Левенштейна между последовательностями p и q, а S' — тестовая выборка, состоящая из пар векторов (x, w). Эту метрику нейронная сети и пытается минимизировать.

Этап декодирования. В [27] было представлено два варианта декодирования интегральных CTC-моделей. Первый метод (нахожде-

ния наилучшего выравнивания выходной последовательности) основывается на предположении, что наиболее вероятное выравнивание соответствует наиболее вероятной выходной последовательности:

$$\arg\max_{w} P(w \mid x) \approx \mathrm{B}(\alpha^*), \tag{12}$$

где $\alpha^* = \arg\max_{\alpha} P(\alpha \mid x)$. Вычисление наилучшего выравнивания является простой задачей, так как α^* — конкатенация наиболее «активных» выходов на каждом временном шаге. Однако это не гарантирует нахождение наиболее вероятной последовательности слов.

Второй метод (метод нахождения префиксов) основывается на факте, что, модифицировав алгоритм прямого-обратного хода, описанный выше, можно эффективно вычислять вероятности последовательных расширений префиксов выходных последовательностей.

Модификации и улучшения. В [30] был предложен метод декодирования, использующий алгоритм лучевого поиска (beam search algorithm), который также позволяет интегрировать языковую модель. Предложенный алгоритм похож на алгоритм декодирования для гибридных СММ/ИНС систем, но отличается интерпретацией выхода нейронной сети. В гибридных системах выходные значения нейронной сети интерпретируются как апостериорные вероятности состояний, которые затем комбинируются с вероятностями перехода и СММ. В CTC сети выходные значения нейронной сети сами представляют собой вероятности перехода. В качестве архитектуры нейронной сети были выбраны двунаправленные LSTM-сети. Сравнивались три модели: RNN-CTC модель, RNN-CTC модель (RNN-WER), переобученная минимизировать WER, и базовая гибридная модель, написанная с помощью инструментария Kaldi [31]. RNN-CTC модель без языковой модели показала WER 30,1%, хотя базовая модель не может быть обучена без ЯМ. Но уже при использовании триграмной ЯМ базовая модель показала WER 7,8%, RNN-CTC — 8,7%, а RNN-WER — 8,2%. Также была протестирована комбинация RNN-CTC и базовой модели, которая показала лучший результат равный 6,7%. В качестве речевого корпуса был использован корпус Wall Street Journal [32].

В [33] и [34] была представлена реализация интегральной системы с использованием инструментария Eesen [33], где декодирование CTC моделей происходило с помощью взвешенных конечных преобразователей (WFST) [35]. Каждый компонент системы: CTC мет-

ки (T), словарь (L) и языковая модель (G) — преобразовывались в один граф поиска следующим образом:

$$TLG = T \circ \min(\det(L \circ G)), \tag{13}$$

где min означает минимизацию, det — детерминацию, и $\circ$ — композицию. Так, *TLG* строит соответствие между последовательностью CTC меток и словами. Это позволяет производить эффективный поиск с помощью, например, библиотеки OpenFST [36]. В [34] были использованы двунаправленные LSTM-сети для распознавания сербской речи. Так был достигнут результат с WER, равной 14,68%, что является не самым хорошим результатом, хотя CER оказалась довольно маленькой — 3,68%.

В [37] была представлена интегральная система с использованием глубоких свёрточных сетей. Также в данной работе была представлена модификация CTC с тремя изменениями: (1) убраны символы «пропуска»; (2) использованы ненормализованные значения в вершинах; (3) применена глобальная нормализация вместо нормализации кадров. Данный метод получил название автоматический критерий сегментации (Auto Segmentation Criterion, ASG). Также для декодирования был использован алгоритм лучевого поиска. По результатам тестов ASG показал более высокую скорость распознавания и меньшую ошибку LER: для CTC на тестовой выборке 10,5%, а ASG — 10,1%. В качестве речевого корпуса был использован корпус LibriSpeech [38]. Система была написана с использованием инструментария Torch7 [39].

В [40] была предложена CTC модель с использованием глубоких свёрточных сетей вместо рекуррентных сетей. Лучшая модель на основе свёрточных сетей имела 10 свёрточных слоев и 3 полносвязных слоя. Лучшая PER оказалась равна 18,2%, при том, что лучшая PER для двунаправленных LSTM сетей оказалась равна 18,3%. Тесты проводились на корпусе TIMIT [41]. Был также сделан вывод, что свёрточные сети позволяют увеличить скорость обучения и больше подходят для обучения на последовательностях фонем.

В [42] были проведены эксперименты по распознаванию речи с использованием CTC моделей на основе LSTM сетей с применением последовательного дискриминантного обучения, а именно минимизация Байесовского риска на уровне состояний (state-level Bayes risk, sMBR) [43]. Эти модели были применены для распознавания детской и взрослой речи с шумом. Также эти модели оказались быстрее в сравнении с комбинацией свёрточных и LSTM сетей, представленных в [44]. Были исследованы два метода для комбинирования моделей:

слияние оценочных метрик (score fusion) и ROVER [45]. Лучший результат был получен с использованием метода ROVER (комбинация двух полносвязных сетей и одной свёрточной) и sMBR, так на тестовых данных взрослой и детской речи была достигнута WER, равная 12,2%. Также были исследованы методы переноса знаний (knowledge transfer) из одной обученной модели в другую и сделан вывод, что это непростая и перспективная задача.

В [46] было проведено исследование интегральных систем с использованием CTC. Было показано, что CTC модель может хорошо работать и без языковых моделей и словаря. В качестве обучающего речевого корпуса был использован корпус [47], составленный из аудиодорожек YouTube видео, общей длительностью более 650 часов. А в качестве тестового корпуса были взяты аудиодорожки из Google Preferred [48] видео, общей длительностью 25 часов. Так, лучшая WER без использования ЯМ была равна 13,9%, а с ЯМ — 13,4%.

Существует «обобщение» CTC моделей — RNN преобразователь (RNN Transducer), который объединяет две RNN в последовательную преобразовательную систему [49, 50]. Одна из сетей похожа на CTC-сеть и обрабатывает тот же момент времени, что и входная последовательность, а вторая RNN моделирует вероятности следующих меток при условии предыдущей. Как и в CTC-сетях, используется динамическое программирование для вычислений и алгоритм прямого-обратного хода, но с учетом ограничений обоих RNN. В отличие от CTC-сетей, RNN преобразователь позволяет генерировать выходные последовательности длиннее входных. RNN преобразователи показали хорошие результаты в распознавании фонем [51] с PER равной 17,7% на корпусе TIMIT.

В [52] было предложено использование глубоких рекуррентных свёрточных сетей и глубоких остаточных сетей совместно с CTC. Лучший результат был получен с применением остаточных сетей с батч-нормализацией. Так был получен результат PER равной 17,33% на речевом корпусе TIMIT.

В [53] были рассмотрены три модели с CTC: ResNet, BLSTM и комбинация LSTM и CNN. Также был предложен метод объединения моделей похожий на ROVER. Так, на речевом корпусе WSJ с помощью ResNet был получен результат — WER, равный 8,99%, а с помощью комбинации трех моделей, упомянутых выше — 7,65%.

Недостатки. CTC модели не лишены недостатков. Во многих работах было отмечено, что при отсутствии языковых моделей CTC-модели часто ошибаются в символах распознанных последовательностей, хотя звучание сохраняется правильным.

Также CTC-модели все еще используют предположение о независимости наблюдаемых переменных. Это значит, что CTC-сети требуется языковая модель, при добавлении которой ошибка распознавания значительно уменьшается [33].

3.4. Инструментарии и библиотеки для построения CTC-моделей. Рассмотрим некоторые примеры инструментариев и библиотек, позволяющих строить системы распознавания речи с использованием CTC.

Keras [54] — это высокоуровневая библиотека для работы с нейронными сетями, написанная на языке Python и использующая инструментарии TensorFlow [55], CNTK или Theano [56]. Так, Keras предоставляет API для использования CTC для обучения нейронных сетей.

CNTK [57] — это открытый инструментарий от Microsoft Research для построения и обучения глубоких нейронных сетей, свёрточных сетей, рекуррентных сетей и сетей с памятью, распространяемый по лицензии MIT. В него была добавлена поддержка CTC и примеры по его использования [58].

Eesen [33] — это легковесная библиотека для построения интегральных систем распознавания речи, использующая рекуррентные сети, CTC и позволяющая выполнять декодирование с помощью WFST или ЯМ на основе рекуррентных нейронных сетей.

Baidu [59] — библиотека, реализующая параллельный алгоритм обучения сетей с использованием CTC. Предоставляет простой интерфейс, написанный на языке C, для использования в различных инструментариях, например TensorFlow, Torch, Theano. Baidu является одной самых быстрых реализаций CTC на данный момент.

Kaldi — свободно распространяемый инструментарий для распознавания речи [31]. Возможности Kaldi позволяют обучать АМ и декодировать модели в системах распознавания речи. С использованием Kaldi и Baidu написана библиотека [60], позволяющая выполнять обучение и декодирование интегральных систем с CTC.

3.5. Шифратор-дешифратор модели, основанные на механизме внимания. Шифратор-дешифратор (Encoder-Decoder) модели часто используются для задач, где длины входной и выходной последовательностей являются переменными [61, 62]. Шифратор (Encoder) — это нейронная сеть, которая трансформирует вход $x = (x_1, \ldots, x_{L'})$ в некоторое промежуточное представление $h = (h_1, \ldots, h_L)$, выделяет признаки. Дешифратор (Decoder) — это обычно RNN, которая использует это промежуточное представление для генерации выходных последовательностей. Шифратор может быть

любой нейронной сетью, например: DNN, LSTM, BLSTM, CNN. На рисунке 5 изображена схема модели.

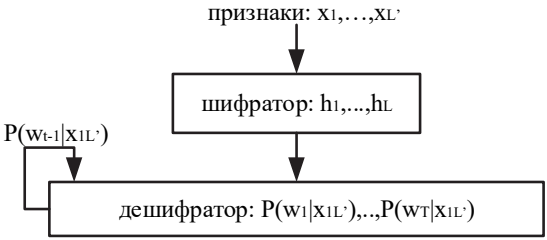

Рис. 5. Шифратор-дешифратор система распознавания речи

В работе [63] в качестве дешифратора было предложено использовать рекуррентный генератор последовательностей, основанный на механизме внимания (Attention-based Recurrent Sequence Generator; ARSG). ARSG — это рекуррентная нейронная сеть, которая стохастически генерирует выходную последовательность $(y_1,\ldots,y_t)$ по входу h длины $L = L'$. ARSG состоит из RNN и из подсети, называемой механизмом внимания (attention-mechanism). Механизм внимания выбирает подпоследовательность входной последовательности, которая затем используется для обновления скрытых состояний RNN и для предсказания следующего выходного значения. На i-ом шаге ARSG генерирует выход y_i, фокусируясь на определенных элементах h:

$$\begin{aligned} \alpha_i &= \text{Attend}(s_{i-1}, \alpha_{i-1}, h) \\ g_i &= \sum_{j=1}^{L} \alpha_{i,j} h_j \\ y_i &= \text{Generate}(s_{i-1}, g_i), \end{aligned} \tag{14}$$

где s_{i-1} — $(i-1)$-е состояние RNN, которое называется Generator (также возможно использование не только RNN), $\alpha_i \in \mathbb{R}^L$ — вектор весов внимания (attention weights), которые также часто называются выравниванием [64]. В [65] g_i было названо «проблеск» (glimpse). Шаг завершается вычислением нового состояния генератора:

$$s_i = \text{Recurrency}(s_{i-1}, g_i, y_i). \tag{15}$$

Recurrency обычно представляет из себя LSTM или GRU [61] модули. Схематично данная модель изображена на рисунке 6.

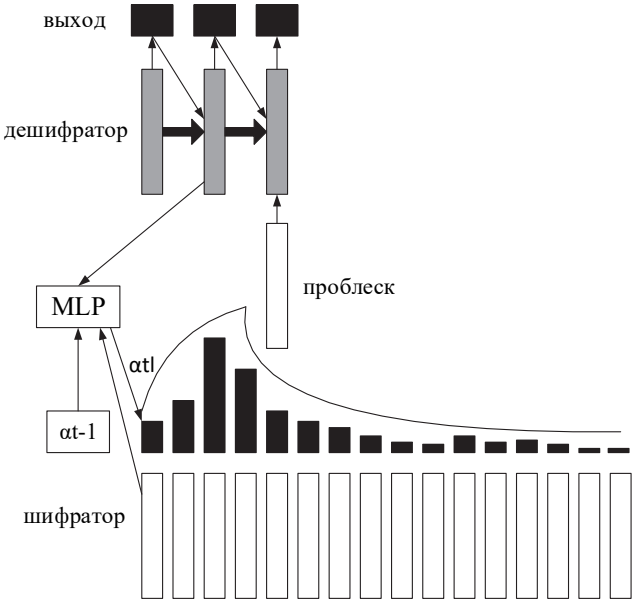

Рис. 6. Интегральная модель, основанная на механизме внимания

Модификации и улучшения. В [63] было предложено разделить механизмы внимания на три вида: по расположению (location-based), по содержанию (content-based) и гибридный, наиболее общий вид. Если Attend не зависит от α_{i-1}, то есть $\alpha_i = \text{Attend}(s_{i-1}, h)$, то это — механизм внимания по содержанию [62]. Attend можно представить как нормализованную сумму метрик каждого элемента h:

$$e_{i,j} = \text{Score}(\text{s}_{i-1}, h_j)$$
$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{j=1}^{L} \exp(e_{i,j})}. \quad (16)$$

Главное ограничение такой схемы в том, что одинаковые или очень похожие элементы h считаются одинаково, несмотря на их позиции в последовательности, что в распознавании речи имеет большое значение. Эта проблема называется «проблемой похожих фрагментов речи». Часто эта проблема частично решается шифратором, например BLSTM или глубокими CNN, которые шифруют контекстную информацию в элементы h. Однако размеры h и их элементов всегда ограничены, что решает данную проблему не в полной мере.

Так, механизм внимания по расположению вычисляет выравнивание с помощью состояния генератора и предыдущего выравнивания, то есть $\alpha_i = \text{Attend}(s_{i-1}, \alpha_{i-1})$. Этот тип механизма внимания предсказывает расстояние между последовательными фонемами или символами только по s_{i-1}, что может быть трудно из-за большой дисперсии этого расстояния.

Гибридный механизм внимания использует предыдущее выравнивание α_{i-1}, чтобы выбрать короткую подпоследовательность h, по которой механизм внимания по содержанию выберет наиболее релевантные элементы без проблемы похожих фрагментов речи.

В [64] была предложена модель с механизмом внимания по содержанию, в которой Score вычисляется следующим образом:

$$e_{i,j} = w^{\mathrm{T}} \tanh(Ws_{i-1} + Vh_j + b), \tag{17}$$

где w и b — вектора, а W и V — матрицы.

В [63] было предложено обобщение этой модели до гибридной. Сначала выделяются k векторов $f_{ij} \in \mathbb{R}^k$ (конволюционные признаки) для каждой позиции j предыдущего выравнивания α_{i-1} с помощью свёртки с матрицей $F \in \mathbb{R}^{k \times r}$:

$$f_i = F * \alpha_{i-1}. \tag{18}$$

Затем вектора f_{ij} используются для операции Score:

$$e_{ij} = w^{\mathrm{T}} \tanh(Ws_{i-1} + Vh_j + Uf_{ij} + b). \tag{19}$$

В формуле (16) есть три проблемы с нормализацией. Во-первых, когда h имеет большую длину, то g_i может содержать много шума из множества незначащих векторов h_j. Во-вторых, механизм внимания должен перебрать все L признаков для каждого y_i для декодирования выходной последовательности длины T, что требует $O(LT)$ операций. Также использование softmax-нормализации в 1 приводит к фокусированию только на одном векторе h_j.

Для решения проблемы шума в g_i применяют заострение (sharpening). В данном методе вводят веса $\beta > 1$:

$$\alpha_{i,j} = \frac{\exp(\beta e_{i,j})}{\sum_{j=1}^{L} \exp(\beta e_{i,j})}. \tag{20}$$

Это позволяет контролировать зашумленные элементы.

Также в [63] для уменьшения количества операции было предложено использование окон (windowing). Для каждого i механизм внимания смотрит только на подпоследовательность $\tilde{h} = (h_{p_i - w},, h_{p_i + w - 1})$ для целой последовательности h, где $w \ll L$ определяет ширину окна, и p_i — медиана выравниваний α_{i-1}. Метрики для $h_j \notin \tilde{h}$ равны 0. Так получаем сложность $O(L + T)$.

Предыдущие техники решают проблемы с длинными последовательностями признаков, но ухудшают работу с обычными признаками. Так, была предложена техника сглаживания (smoothing) уравнения 20. Неограниченная функция экспоненты заменяется на ограниченную сигмоидную функцию σ:

$$\alpha_{i,j} = \frac{\sigma(e_{i,j})}{\sum_{j=1}^{L} \sigma(e_{i,j})}. \tag{21}$$

В [66] также была предложена интеграция модели, основанной на механизме внимания, и ЯМ. Для того чтобы построить ЯМ, основанную на символах, из модели, построенной на словах, использовались WFST. Так, был построен конечный автомат $T = \min(\det(L \circ G))$, где L — словарь и G — конечный автомат для ЯМ. При декодировании запускался поиск выхода y, который минимизировал функционал L, комбинирующий модель шифратор-дешифратор и ЯМ:

$$L = -\log P_{ED}(y \mid x) - \beta \log P_{LM}(y) - \gamma T, \tag{22}$$

где β и γ — настраиваемые параметры. В итоге на речевом корпусе WSJ были получены следующие результаты: WER, равная 11,3%, и CER, равная 4,8%.

В [67] независимо была предложена похожая модель, основанная на механизме внимания, названная «Listen, Attend and Spell» (LAS). Шифратор представлял собой BLSTM пирамидальной структуры, а дешифратор использовал LSTM. Также полученная модель после декодирования пересчитывалась с помощью ЯМ. Так на речевом корпусе Google Voice Search была получена WER, равная 10,3%.

В [68] была предложена модель, объединяющая CTC и модели, основанной на механизме внимания. Идея данной модели в том, чтобы использовать CTC функцию для обучения шифратора модели. Так, на

чистом речевом корпусе WSJ1 были получены результаты: WER равная 18.2% и CER равная 7,36%.

В [69] были предложены различные техники, такие как: монотонная регуляризация (monotonic regularization); плановое обучение (Curriculum learning) [70], когда длины входных последовательностей увеличиваются с обучением; «плоский старт» (flatstart) [42] — для выбора начальных позиций в зависимости от темпа речи диктора.

В [71] была рассмотрен подход к модификации LAS системы с использованием свёрточных LSTM сетей с остаточными модулями и батчнормализацией. Так, лучший результат на речевом корпусе WSJ по показателю WER составил 10,53%.

В работе [72] был описан подход обучения модели, основанной на механизме внимания, с использованием необработанных звуковых признаков и техники переноса знаний (transfer learning). Для упрощения обучения на стадии шифрования модель имела следующую архитектуру: нижние слои шифратора состояли из нескольких свёрточных и обычных слоев, предсказывающих спектральные признаки по необработанным данным, а именно мелчастотные кепстральные коэффициенты (MFCC) и коэффициенты перцептивного линейного предсказания (log Mel-scale spectrogram). Лучший результат на корпусе WLJ по показателю CER был равен 14,71%.

3.6. Инструментарии и библиотеки для построения шифратор-дешифратор моделей на основе механизма внимания. Рассмотрим примеры библиотек и инструментариев, позволяющих реализовать модели на основе механизма внимания.

Theano+Bricks+Fuel — это инструментарий [73], написанный с помощью библиотек Theano, Bricks и Fuel [74] и использовавшийся в [64, 75].

Tensor2Tensor — популярная библиотека, написанная с использованием TensorFlow и позволяющая строить обобщенные модели. Библиотека предоставляет возможность использовать модели, основанные на механизме внимания [76].

Keras — с помощью инструментария Keras была написана библиотека для построения моделей шифратор-дешифратор и моделей, основанных на механизме внимания [77].

3.7. Условные случайные поля. В [78] был предложен еще один метод для вычисления условных вероятностных распределений, которые можно использовать для распознавания речи — условные случайные поля (Conditional Random Field; CRF).

Эту модель определяют следующим образом. Пусть X — множество последовательностей, которые нужно распознать, а Y — множество последовательностей меток над алфавитом Υ. Необходимо

построить распределение $P(Y \mid X)$. Пусть $G = (V, E)$ — граф, где $Y = (Y_v)_{v \in V}$, так что Y индексировано вершинами графа G. Тогда (X, Y) называется условным случайным полем (CRF), если $P(Y_v \mid X, Y_w, w \neq v) = P(Y_v \mid X, Y_w, w \sim v)$. Так, CRF представляет собой неориентированный граф G, где каждая вершина является случайной переменной и каждое ребро представляет собой зависимость между случайными переменными. На рисунке 7 представлена схема линейного условного случайного поля. Пустые кружки означают, что соответствующая переменная не генерируется моделью.

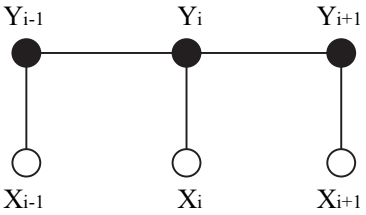

Рис. 7. Линейное условное случайное поле для последовательностей

Для вычисления условной вероятности $P(Y \mid X)$ можно определить набор потенциальных функций (potential function) φ для каждой клики графа $c \in C$, где C — множество клик графа G (клика — полный подграф неориентированного графа). Функция φ каждому возможному состоянию элементов клики ставит в соответствие некоторое неотрицательное число. Вершины, не являющиеся смежными, должны соответствовать условно независимым случайным величинам. Группа смежных вершин образует клику, набор состояний вершин является аргументом соответствующей потенциальной функции. Обозначим y_c множество случайных переменных из Y, соответствующих клике c. Тогда:

$$P(y \mid X) = \frac{1}{Z(X)} \prod_{c \in C} \varphi(y_c, X), \tag{23}$$

где $Z(X)$ — нормализующий коэффициент:

$$Z(X) = \sum_{y' \in Y'} \prod_{c \in C} \varphi(y'_c, X). \tag{24}$$

Обозначим множество переменных, соответствующих клике c, в момент времени t как $y_{t,c}$, тогда можно вычислять $P(Y \mid X)$ следующим образом:

$$P(y \mid X) = \frac{1}{Z(X)} \prod_{c \in C} \prod_{t=1}^{T} \exp(\lambda_c^{\mathrm{T}} \cdot f(y_{t,c}, X)), \tag{25}$$

где λ_c — множество параметров для клики c.

Метод CRF, как и метод MEMM (Maximum Entropy Markov Models) [79], относится к дискриминативным вероятностным методам, в отличие от генеративных методов, таких как CMM.

По аналогии с MEMM, выбор признаков для задания вероятности перехода между состояниями при наличии наблюдаемого значения зависит от данных. Но в отличие от MEMM, CRF может учитывать любые особенности и взаимозависимости в исходных данных. Вектор признаков рассчитывается на основе обучающей выборки и определяет вес каждой потенциальной функции. Для обучения и применения модели используются алгоритмы, аналогичные алгоритмам CMM: Витерби и его разновидность — алгоритм прямого-обратного хода (forward-backward algorithm).

CMM можно рассматривать как частный случай линейного условного случайного поля (linear-chain CRF). В условных случайных полях отсутствует так называемая проблема смещения меток (label bias problem) [80] — ситуация, когда преимущество получают состояния с меньшим количеством переходов, так как строится единое распределение вероятностей и нормализация производится в целом, а не в рамках отдельного состояния. Так, алгоритм не требует предположения независимости наблюдаемых переменных.

Недостатки. Недостатком подхода CRF является вычислительная сложность анализа обучающей выборки, что затрудняет постоянное обновление модели при поступлении новых обучающих данных.

Модификации и улучшения. В [81] было предложено использовать модификацию CRF — дополненные CRF (Augmented CRF; ACRF). На речевом корпусе TIMIT был получен результат по показателю PER, равный 23,0%.

В [82] были предложены сегментные рекуррентные нейронные сети (Segmental Recurrent Neural Networks; SRNN). Они строятся на

основе модификации CRF — сегментных условных случайных полей (semi-Markov CRF), которые можно описать как:

$$P(y, E \mid X) = \frac{1}{Z(X)} \prod_{t=1}^{T} \exp(f(y_t, e_t, X)), \tag{26}$$

где $E = (e_1, \ldots, e_T)$ — вспомогательные сегментные метки, $e_t = \langle s_t, n_t \rangle$ — пара из начала s_t и конца n_t временной метки сегмента y_t, а $Z(X)$ — нормализующий коэффициент:

$$Z(X) = \sum_{y,E} \prod_{t=1}^{T} \exp(f(y_t, e_t, X)). \tag{27}$$

Функция $f(\cdot)$ определяется следующим образом:

$$f(y_t, e_t, X) = w^{\mathrm{T}} \Phi(y_t, e_t, X), \tag{28}$$

где $\Phi(\cdot)$ — функция признаков, и w — вектор весов.

Для определения $\Phi(\cdot)$ используют рекуррентные нейронные сети, а параметры, в частности E, определяют с помощью модификации функции потерь на основе максимального правдоподобия. Так, сначала y_t представляются в виде прямого унитарного кода v_t, а затем переводится в непрерывное пространство с помощью матрицы M, определяющей векторное представление меток (embedding matrix):

$$u_t = M v_t. \tag{29}$$

Для отображения акустических сегментов в вектора фиксированного размера используется рекуррентная нейронная сеть:

$$\begin{aligned} h_1^t &= r(h_0, x_{s_t}) \\ h_2^t &= r(h_1^t, x_{s_t+1}) \\ h_{d_t}^t &= r(h_{d_t-1}^t, x_{n_t}), \end{aligned} \tag{30}$$

где h_0 означает начальное скрытое состояние сети, $d_t = n_t - s_t$ — длины сегмента, и $r(\cdot)$ — нелинейную функцию. Так,

$$\Phi(y_t, e_t, X) = g(u_t, h_{d_t}^t), \tag{31}$$

где $g(\cdot)$ соответствует одному или нескольким слоям линейных и нелинейных преобразований. На рисунке 8 представлена схема сегментной рекуррентной сети с CRF, где закрашенные кружки обозначают $h_{d_t}^t$.

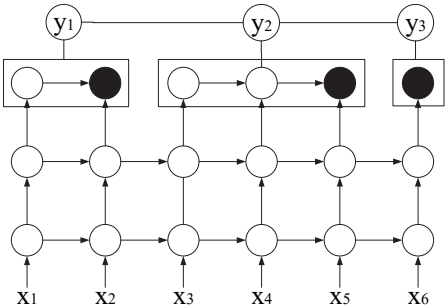

Рис. 8. Сегментные рекуррентные нейронные сети использующие CRF первого порядка

В [83] SRNN были применены для распознавания речи. На речевом корпусе TIMIT был получен результат по показателю PER, равный 17,3%. Предложенная модель не использовала ЯМ.

3.8. Сравнение результатов. В таблице 1 представлены результаты применения рассмотренных методов (гибридные СММ/ИНС модели, CTC-модели, модели, на основе механизма внимания и CRF-модели).

Таблица 1. Сравнение результатов

Модель	ЯМ	Технологии	Тестовый речевой корпус	WER, %	CER, %	PER, %
Гибридные СММ/ИНС модели						
CNN [84]	✓	Torch7	WSJ (Nov'92)	6,7	–	–
BLSTM [13]	✓	–	TIMIT	–	–	17,99
CLDNN-HMM [44]	✓	–	Google Voice Search	8,0	–	–
Kaldi-dnn5b-pretrain-dbn-dnn-smbr recipe [31]	✓	Kaldi	WSJ (Nov'92)	3,35	–	–
CTC-модели						
RNN-CTC + Kaldi + trigram LM [30]	✓	Kaldi	WSJ (Nov'92)	6,7	–	–
LSTM-CTC для фонем + trigram LM [33]	✓	Eesen	WSJ (Nov'92)	7,9	–	–

Продолжение таблицы 1

Шифратор-дешифратор модели, на основе механизма внимания						
LSTM-CTC + trigram LM [33]	✓	Eesen	WSJ (Nov'92)	7,3	–	–
RCNN + BLSTM + CLDNN + CTC [53]	✓	–	WSJ (Nov'92)	7,65	–	–
CNN + RNN + CTC [70]	✓	Baidu	WSJ (Nov'92)	4,42	–	–
BLSTM-CTC [34]	✓	Eesen	Корпус сербской речи	14,7	3,7	–
CNN + ASG [37]	✓	Torch7, Baidu	LibriSpeech	7,2	–	–
ROVER: LSTM + CNN + sMBR [42]	✓	–	Google Now+Youtube Kids	12,2	–	–
BLSTM + CTC + LM [46]	✓	TensorFlow	YouTube video	13,4	–	–
CNN + CTC [40]	×	Theano, Blocks, Fuel	TIMIT	–	–	18,2
ResNet + CTC [52]	✓	Lasagne [86], Baidu	TIMIT	–	–	17,33
RNN Transducer [51]	×	–	TIMIT	–	–	17,7
ARSG + конв. признаки + сглаживание [63]	×	Theano, PyLearn2, Blocks	TIMIT	–	–	17,6
ARSG + trigram LM [66]	✓	Theano, Blocks, Fuel	WSJ (Nov'92)	9,3	3,9	–
ARSG + CTC [68]	×	Chainer [87]	WSJ (Nov'92)	18,2	7,36	–
LAS + LM [67]	✓	DistBelief [88]	Google Voice Search	10,3	–	–
LAS + CNN + LSTM + ResNet [71]	×	TensorFlow	WSJ (Nov'92)	10,5	–	–
Att. + transfer learning [72]	×	PyTorch	WSJ (Nov'92)	17,04	14,71	–
CRF-модели						
SRNN [83], [89]	×	Kaldi, DyNet [90]	TIMIT	–	–	17,3

Как можно видеть из таблицы, интегральные системы в настоящее время немного уступают гибридным моделям по точности распознавания. Но заметим, что CTC-модели являются наиболее простыми с точки зрения архитектуры и при условии использования языковых моделей дают близкие к гибридным моделям результаты. Во многих работах было отмечено, что CTC-модели часто ошибаются в символах распознанных последовательностей, хотя звучание сохраняется правильным. Именно этот недостаток и заставляет использовать отдельно обученные языковые модели.

Стоит отметить, что шифратор-дешифратор архитектуры на основе механизма внимания также показывают перспективные результаты, так как даже без применения языковых моделей демонстрируют низкую погрешность распознавания. А также при малом размере обучающей выборки применение техники переноса «знаний» с модели, обученной на другом языке, является перспективным направлением, которое может быть использовано для создания универсальных систем распознавания речи или систем для редких языков.

Рассмотрим отличия интегральных моделей друг от друга, их преимущества и недостатки. В работах было отмечено, что CTC-модели позволяют достичь хороших результатов только при использовании языковых моделей, но была показана относительная простота их реализации и обучения по сравнению с шифратор-дешифратор моделями, которые, как было показано в работах, позволяют достичь приемлемых результатов и без использования языковых моделей. При этом была отмечена сложность процесса обучения шифратор-дешифратор моделей по причине большого числа гиперпараметров нейронной сети и необходимость объемного речевого корпуса, что может представлять собой проблему для малоиспользуемых языков. Использование CRF-моделей является пока что развивающейся областью и не продемонстрировало достаточных результатов для сравнения.

3.9. Сравнение инструментариев для построения интегральных моделей. В таблице 2 представлено сравнение библиотек и инструментариев, позволяющих создавать и обучать интегральные модели для распознавания речи, которые были рассмотрены в данном обзоре.

Таблица 1. Сравнение инструментариев

Название	Модели			ЯМ	Платформа	Язык	API
	a	b	c				
TensorFlow	✓	✓		✓	Linux, OS X, Windows	C++, Python	C++, Python, Java, Haskell
Kaldi	✓	✓	✓	✓	Linux, Windows	C++, bash	C++, bash
Eesen	✓	✓		✓	Linux	C++	C++, bash, Python
Baidu	✓	×		×	Linux, OS X	C++	C++, Python
Torch	✓	✓		✓	Linux, OS X, Wndows	C, Lua	Lua, C
Theano	✓	✓		✓	Linux, OS X, Windows	C++, Python	Python, C++
Chainer		×		×	Linux	Python	Python
CNTK	✓	×		×	Linux, Windows	C++, Python	Python, C++, C#, Java

Модели (a), (b) и (c) означают CTC-модели, шифратор-дешифратор и CRF-модели соответственно. Галочка стоит напротив тех инструментариев, которые были использованы в рассмотренных выше статьях для построения соответствующих моделей.

Очевидно, что с помощью данных библиотек можно реализовать практически любые модели, но информация о применении была взята из официальных репозиториев с примерами из статей.

4. Заключение. В данном обзоре были рассмотрены основные методы построения интегральных моделей распознавания речи, такие как: CTC-модели, модели на основе механизма внимания и CRF-модели. Как можно видеть из таблицы 1, интегральные системы пока что немного уступают в точности распознавания гибридным СММ/ИНС моделям. Но можно отметить такие преимущества интегральных систем, как возможность устранения «тяжелых» языковых моделей, упрощение системы и более быстрая работа по сравнению с гибридными СММ/ИНС моделями. В перспективе интегральные модели предоставляют возможность качественного распознавания слитной речи на мобильных устройствах локально без обработки сигнала на удаленных серверах, при этом используя меньше памяти и вычислительных мощностей, чем гибридные модели.

Литература

1. *Ронжин А.Л., Карпов А.А., Ли И.В.* Речевой и многомодальный интерфейсы // М.: Наука. 2006. 173 с.
2. *Ganchev T., Fakotakis N., Kokkinakis G.* Comparative evaluation of various MFCC implementations on the speaker verification task // Proceedings of the SPECOM. 2005. pp. 191–194.
3. *Hermansky H., Malayath N.* Speaker verification using speaker-specific mappings // Proc. RLA2C. 1998. 4 p.
4. *Маковкин К.А.* Гибридные модели – Скрытые марковские модели. Многослойный персептрон и их применение в системах распознавания речи. Обзор // Речевые технологии. 2012. № 3. С. 58–83.
5. *Cosi P.* A KALDI-DNN-based ASR system for Italian // 2015 International Joint Conference on Neural Networks (IJCNN). 2015. pp. 1–5.
6. *Kipyatkova I.*, *Karpov A.* DNN-Based Acoustic Modeling for Russian Speech Recognition Using Kaldi // International Conference on Speech and Computer. 2016. pp. 246–253.
7. *LeCun Y., Bengio Y.* Convolutional networks for images, speech, and time series // The handbook of brain theory and neural networks. 1995. vol. 3361. no. 10. pp. 1995.
8. *Abdel-Hamid O. et al.* Convolutional neural networks for speech recognition // IEEE/ACM Transactions on audio, speech, and language processing. 2014. vol. 22. no. 10. pp. 1533–1545.
9. *Sainath T.N., Mohamed A.-R., Kingsbury B., Ramabhadran B.* Deep convolutional neural networks for LVCSR // 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2013. pp. 8614–8618.
10. *Robinson T., Hochberg M., Renals S.* The use of recurrent neural networks in continuous speech recognition // Automatic speech and speaker recognition. 1996. pp. 233–258.

11. *Hochreiter S., Schmidhuber J.* Long short-term memory // Neural computation. 1997. vol. 9. no. 8. pp. 1735–1780.
12. *Гапочкин А.В.* Нейронные сети в системах распознавания речи // Science Time. 2014. № 1(1). pp. 29–36.
13. *Graves A., Jaitly N., Mohamed A.-R.* Hybrid speech recognition with deep bidirectional LSTM // 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). 2013. pp. 273–278.
14. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 770–778.
15. *Markovnikov N.M., Kipyatkova I., Karpov A., Filchenkov A.* Deep neural networks in Russian speech recognition // Proceedings of 2017 Artificial Intelligence and Natural Language Conference. 2017. pp. 54–67.
16. *Кипяткова И.С., Карпов А.А.* Разновидности глубоких искусственных нейронных сетей для систем распознавания речи // Труды СПИИРАН. 2016. Вып. 49(6). С. 80–103.
17. *Ackley D.H., Hinton G.E., Sejnowski T.J.* A learning algorithm for Boltzmann machines // Cognitive science. 1985. vol. 9. no. 1. pp. 147–169.
18. *Srivastava N. et al.* Dropout: a simple way to prevent neural networks from overfitting // Journal of machine learning research. 2014. vol. 15. no. 1. pp. 1929–1958.
19. *Ioffe S., Szegedy C.* Batch normalization: Accelerating deep network training by reducing internal covariate shift // International Conference on Machine Learning. 2015. pp. 448–456.
20. *Levenshtein V.I.* Binary codes capable of correcting deletions, insertions, and reversals // Soviet physics. Doklady. 1996. vol. 10. pp. 707–710.
21. *Mikolov T., et al.* Recurrent neural network based language model // Interspeech. 2010. vol. 2. pp. 1045–1048.
22. *Rao K., Peng F., Sak H., Beaufays F.* Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks // 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. pp. 4225–4229.
23. *Jaitly N., Hinton G*. Learning a better representation of speech soundwaves using restricted boltzmann machines // 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2011. pp. 5884–5887.
24. *Smolensky P.* Information processing in dynamical systems: Foundations of harmony theory // Colorado University at Boulder Dept of Computer Science. 1986. pp. 194–281.
25. *Bojarski M. et al.* End to End Learning for Self-Driving Cars» // 2016. preprint: arXiv: 1604.07316. URL: https://arxiv.org/abs/1604.07316 (дата обращения 17.02.2018).
26. *Sayre K.M.* Machine recognition of handwritten words: A project report // Pattern recognition. 1973. vol. 5. no. 3. pp. 213–228.
27. *Graves A., Ferna´ndez S., Gomez F., Schmidhuber J.* Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks // Proceedings of the 23rd international conference on Machine learning. 2006. pp. 369–376.
28. *Bridle J.S.* Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition // Neurocomputing. 1990. pp. 227–236.
29. *Rabiner L.R.* A tutorial on hidden Markov models and selected applications in speech recognition // Proceedings of the IEEE. 1989. vol. 77. no. 2. pp. 257–286.
30. *Graves A., Jaitly N.* Towards end-to-end speech recognition with recurrent neural networks // Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014. pp. 1764–1772.
31. *Povey D. et al.* The Kaldi speech recognition toolkit» // IEEE 2011 workshop on automatic speech recognition and understanding. IEEE Signal Processing Society. 2011. 4 p.
32. Корпус английской речи WSJ. URL: https://catalog.ldc.upenn.edu/LDC93S6B (дата обращения: 17.02.2018).

33. *Miao Y., Gowayyed M., Metze F.* EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding // 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). 2015. pp. 167–174.
34. *Popovi´c B., Pakoci E., Pekar D.* End-to-End Large Vocabulary Speech Recognition for the Serbian Language // International Conference on Speech and Computer. 2017. pp. 343–352.
35. *Mohri M., Pereira F., Riley M.* Weighted finite-state transducers in speech recognition // Computer Speech & Language. 2002. vol. 16. no. 1. pp. 69–88.
36. *Allauzen C. et al.* A general and efficient weighted finite-state transducer library // International Conference on Implementation and Application of Automata. 2007. pp. 11–23.
37. *Collobert R., Puhrsch C., Synnaeve G.* Wav2letter: an end-to-end convnetbased speech recognition system // 2016. preprint: arXiv: 1609.03193. URL: https://arxiv.org/abs/1609.03193 (дата обращения 17.02.2018).
38. *Panayotov V., Chen G., Povey D., Khudanpur S.* Librispeech: an ASR corpus based on public domain audio books // 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. pp. 5206–5210.
39. Deep learning toolkit Torch. URL: http://www.torch.ch/ (дата обращения 17.02.2018).
40. *Zhang Y. et al.* Towards end-to-end speech recognition with deep convolutional neural networks // 2017. preprint: arXiv: 1701.02720. URL: https://arxiv.org/abs/1701.02720 (дата обращения 17.02.2018).
41. Корпус английской речи TIMIT. URL: https://catalog.ldc.upenn.edu/ldc93s1 (дата обращения: 17.02.2018).
42. *Sak H., de Chaumont Quitry F., Sainath T., Rao K.* Acoustic modelling with cd-ctc-smbr lstm rnns // 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). 2015. pp. 604–609.
43. *Kingsbury B.* Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling // 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009). 2009. pp. 3761–3764.
44. *Sainath T.N., Vinyals O., Senior A., Sak H.* Convolutional, long shortterm memory, fully connected deep neural networks // 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. pp. 4580–4584.
45. *Fiscus J.G.* A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER) // Proceedings of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding. 1997. pp. 347–354.
46. *Soltau H., Liao H., Sak H.* Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition // 2016. preprint: arXiv: 1610. 09975. URL: https://arxiv.org/abs/1610.09975 (дата обращения 17.02.2018).
47. *Liao H., McDermott E., Senior A.* Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription // 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). 2013. pp. 368–373.
48. Youtube. URL: https://www.youtube.com/yt/lineups/ (дата обращения 17.02.2018).
49. *Graves A.* Sequence transduction with recurrent neural networks // 2012. preprint: arXiv: 1211.3711. URL: https://arxiv.org/abs/1211.3711 (дата обращения 17.02.2018).
50. *Boulanger-Lewandowski N., Bengio Y., Vincent P.* High-dimensional sequence transduction // 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2013. pp. 3178–3182.
51. *Graves A., Mohamed A.-R., Hinton G.* Speech recognition with deep recurrent neural networks // 2013 IEEE International Conference on Acoustics, speech and signal processing (ICASSP). 2013. pp. 6645–6649.
52. *Zhang Z. et al.* Deep Recurrent Convolutional Neural Network: Improving Performance For Speech Recognition // 2016. preprint: arXiv: 1611.07174. URL: https://arxiv.org/abs/1611.07174 (дата обращения 17.02.2018).
53. *Wang Y., Deng X., Pu S., Huang Z.* Residual convolutional CTC networks for automatic speech recognition // 2017. preprint: arXiv: 1702.07793. URL: https://arxiv.org/abs/1702.07793 (дата обращения 17.02.2018).

54. Keras: The Python Deep Learning library. URL: https://keras.io/ (дата обращения 17.02.2018).
55. TensorFlow. An open source machine learning framework for everyone. URL: https://www.tensorflow.org/ (дата обращения 17.02.2018).
56. Инструментарий для глубокого обучения Theano. URL: http://deeplearning.net/software/theano/ (дата обращения 17.02.2018).
57. The Microsoft Cognitive Toolkit. URL: https://docs.microsoft.com/ru-ru/cognitive-toolkit/ (дата обращения 17.02.2018).
58. Example implementation of speech recognition system. URL: https://github.com/Microsoft/CNTK/tree/master/Tests/EndToEndTests/Speech/LSTM_CTC_MLF (дата обращения 17.02.2018).
59. CTC loss-function implementation. URL: https://github.com/baidu-research/warp-ctc (дата обращения 17.02.2018).
60. CTC model implementation using Kaldi. URL: https://github.com/lingochamp/kaldi-ctc (дата обращения 17.02.2018).
61. *Cho K. et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation // 2014. preprint: arXiv: 1406.1078. URL: https://arxiv.org/abs/1406.1078 (дата обращения 17.02.2018).
62. *Sutskever I., Vinyals O., Le Q.V.* Sequence to sequence learning with neural networks // Advances in neural information processing systems. 2014. pp. 3104–3112.
63. *Chorowski J.K. et al.* Attentionbased models for speech recognition // Advances in Neural Information Processing Systems. 2015. pp. 577–585.
64. *Bahdanau D., Cho K., Bengio Y.* Neural machine translation by jointly learning to align and translate // 2014. preprint: arXiv: 1409.0473. URL: https://arxiv.org/abs/1409.0473 (дата обращения 17.02.2018).
65. *Mnih V., Heess N., Graves A.* Recurrent models of visual attention // Advances in neural information processing systems. 2014. pp. 2204–2212.
66. *Bahdanau D. et al.* End-to-end attention-based large vocabulary speech recognition // 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016. pp. 4945–4949.
67. *Chan W., Jaitly N., Le Q., Vinyals O.* Listen, attend and spell: A neural network for large vocabulary conversational speech recognition // 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016. pp. 4960–4964.
68. *Kim S., Hori T., Watanabe S.* Joint CTC-attention based end-to-end speech recognition using multi-task learning // 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2017. pp. 4835–4839.
69. *Chorowski J., Bahdanau D., Cho K., Bengio Y.* End-to-end continuous speech recognition using attention-based recurrent NN: first results // 2014. preprint: arXiv: 1412.1602. URL: https://arxiv.org/abs/1412.1602 (дата обращения 17.02.2018).
70. *Amodei D. et al.* End to end speech recognition in English and Mandarin // ICLR 2016 workshop. 2016. 12 p.
71. *Zhang Y., Chan W., Jaitly N.* Very deep convolutional networks for end-to-end speech recognition // 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2017. pp. 4845–4849.
72. *Tjandra A., Sakti S., Nakamura S.* Attention-based Wav2Text with feature transfer learning // 2017. preprint: arXiv: 1709.07814. URL: https://arxiv.org/abs/1709.07814 (дата обращения 17.02.2018).
73. Implementation of attention-based model. URL: https://github.com/rizar/attention-lvcsr (дата обращения 17.02.2018).
74. *Van Merri¨enboer et al.* Blocks and fuel: Frameworks for deep learning // 2015. preprint: arXiv: 1506.00619. URL: https://arxiv.org/abs/1506.00619. (дата обращения 17.02.2018).
75. *Bahdanau D. et al.* Task loss estimation for sequence prediction // 2015. preprint: arXiv: 1511.06456. URL: https://arxiv.org/abs/1511.06456 (дата обращения 17.02.2018).
76. Luong T., Brevdo E., Zhao R. Neural machine translation (seq2seq) tutorial. 2017. URL: https://www.tensorflow.org/tutorials/seq2seq (дата обращения 17.02.2018).

77. Implementation of end-to-end models. URL: https://github.com/farizrahman4u/seq2seq (дата обращения 17.02.2018).
78. *Lafferty J., McCallum A., Pereira F.C.* Conditional random fields: Probabilistic models for segmenting and labeling sequence data // 2001. 8 p.
79. *Fosler-Lussier E., He Y., Jyothi P., Prabhavalkar R.* Conditional random fields in speech, audio, and language processing // Proceedings of the IEEE. 2013. vol. 101. no. 5. pp. 1054–1075.
80. *Bottou L.* Une Approche th´eorique de l'Apprentissage Connexioniste; Applications `a la reconnaissance de la Parole // Ph.D. thesis. Universite de Paris XI. 1991. 236 p.
81. *Hifny Y., Renals S.* Speech recognition using augmented conditional random fields // IEEE Transactions on Audio, Speech, and Language Processing. 2009. vol. 17. no. 2. pp. 354–365.
82. *Kong L., Dyer C., Smith N.A.* Segmental recurrent neural networks // 2015. preprint: arXiv: 1511.06018. URL: https://arxiv.org/abs/1511.06018 (дата обращения 17.02.2018).
83. *Lu L., et al.* Segmental recurrent neural networks for end-to-end speech recognition // 2016. preprint: arXiv: 1603.00223. URL: https://arxiv.org/abs/1603.00223 (дата обращения 17.02.2018).
84. *Palaz D., Doss M.M., Collobert R.* Convolutional neural networksbased continuous speech recognition using raw speech signal // 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. pp. 4295–4299.
85. *Amodei D. et al.* Deep speech 2: End-to-end speech recognition in english and mandarin // International Conference on Machine Learning. 2016. pp. 173–182.
86. Deep learning toolkit Lasagne. URL: http://lasagne.readthedocs.io/en/latest/ (дата обращения 17.02.2018).
87. Chainer. A Powerful, Flexible, and Intuitive Framework for Neural Networks. URL: https://chainer.org/ (дата обращения 17.02.2018).
88. *Dean J. et al.* Large scale distributed deep networks // Advances in neural information processing systems. 2012. pp. 1223–1231.
89. *Lu L., Kong L., Dyer C., Smith N.A.* Multi-task Learning with CTC and Segmental CRF for Speech Recognition // 2017. preprint: arXiv: 1702.06378. URL: https://arxiv.org/abs/1702.06378 (дата обращения 17.02.2018).
90. *Neubig G. et al.* DyNet: The Dynamic Neural Network Toolkit // 2017. preprint: arXiv: 1701.03980. URL: https://arxiv.org/abs/1701.03980 (дата обращения 17.02.2018).

Марковников Никита Михайлович — программист лаборатории речевых и многомодальных интерфейсов, Федеральное государственное бюджетное учреждение науки Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: распознавание речи, нейронные сети, глубокое обучение. Число научных публикаций — 1. niklemark@gmail.com; 14-я линия В.О., 39, Санкт-Петербург, 199178; р.т.: +7(812)328-3337, Факс: +7(812)328-4450.

Кипяткова Ирина Сергеевна — к-т техн. наук, старший научный сотрудник лаборатории речевых и многомодальных интерфейсов, Федеральное государственное бюджетное учреждение науки Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН), доцент кафедры управления в технических системах, Санкт-Петербургский государственный университет аэрокосмического приборостроения (СПбГУАП). Область научных интересов: автоматическое распознавание речи, статистические модели языка, нейронные сети. Число научных публикаций — 75. kipyatkova@iias.spb.su; 14-я линия В.О., 39, Санкт-Петербург, 199178; р.т.: +7(812)328-0421, Факс: +7(812)328-0421.

Поддержка исследований. Работа выполнена при финансовой поддержке фонда РФФИ (проекты № 18-07-01216 и 18-07-01407), Совета по грантам Президента РФ (проекты № МК-1000.2017.8 и МД-254.2017.8) и бюджетной темы № 0073-2018-0002.

DOI 10.15622/sp.58.4

N.M. MARKOVNIKOV, I.S. KIPYATKOVA

AN ANALYTIC SURVEY OF END-TO-END SPEECH RECOGNITION SYSTEMS

Markovnikov N.M., Kipyatkova I.S. **An Analytic Survey of End-to-End Speech Recognition Systems.**

Abstract. This article presents an analytic survey of various end-to-end speech recognition systems, as well as some approaches to their construction, training and optimization. We consider models based on connectionist temporal classification (CTC) as a loss function for neural networks, models based on encoder-decoder architecture with attention mechanism. Also, we describe neural networks models built using conditional random field (CRF), that is a generalization of hidden markov models that allows to fix some drawbacks of standard hybrid speech recognition systems like an assumption of independency of elements from speech frames sequences. We also describe integration possibilities with language models at a stage of decoding for end-to-end systems. Also, various modification and improvements of standard end-to-end models, for example, like generalization of connectionist temporal classification and regularization using at attention-based encoder-decoder models. We see that such an approach significantly reduces recognition error rates for end-to-end models. A survey of research works in this subject area reveals that end-to-end systems allow achieving results close to that of the state-of-the-art hybrid models. Nevertheless, end-to-end models use simple configuration and demonstrate a high speed of learning and decoding. In addition, we consider popular frameworks and toolkits for creating speech recognition systems like TensorFlow, Eesen, Kaldi, etc. Theirs comparing was provided by simplicity and accessibility of implementation end-to-end speech recognition system.

Keywords: speech recognition, end-to-end models, neural networks, deep learning.

Markovnikov Nikita Mikhailovich — programmer of speech and multimodal interfaces laboratory , St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: speech recognition, neural networks, deep learning. The number of publications — 1. niklemark@gmail.com; 39, 14-th Line V.O., St.-Petersburg, 199178, Russia; office phone: +7(812)328-3337, Fax: +7(812)328-4450.

Kipyatkova Irina Sergeevna — Ph.D., senior researcher of speech and multimodal interfaces laboratory , St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), associate professor of control in technical systems department, Saint Petersburg State University of Aerospace Instrumentation (SUAI). Research interests: automatic speech recognition, statistical language models. The number of publications — 75. kipyatkova@iias.spb.su; 39, 14-th Line V.O., St.-Petersburg, 199178, Russia; office phone: +7(812)328-0421, Fax: +7(812)328-0421.

Acknowledgements. This research is supported by the Russian Foundation for Basic Research (projects No. 18-07-01216 and 18-07-01407), by the Council for Grants of the President of the Russian Federation (projects No. MK-1000.2017.8 and MD-254.2017.8) and state research № 0073-2018-0002.

References

1. Ronzhin A.L., Karpov A.A., Li I.V. *Rechevoj i mnogomodal'nyj interfejsy* [Speech and multimodal interfaces]. M.: Nauka. 2006. 173 p. (In Russ.).
2. Ganchev T., Fakotakis N., Kokkinakis G. Comparative evaluation of various MFCC implementations on the speaker verification task. Proceedings of the SPECOM. 2005. pp. 191–194.

3. Hermansky H., Malayath N. Speaker verification using speaker-specific mappings. Proc. RLA2C. 1998. 4 p.
4. Makovkin K.A. [Hybrid models – Hidden Markov Models/Multilayer perceptron and their application in speech recognition systems. Servey]. *Rechevye tehnologii – Speech Technology*. 2012. vol. 3. pp. 58–83. (In Russ.).
5. Cosi P. A KALDI-DNN-based ASR system for Italian. 2015 International Joint Conference on Neural Networks (IJCNN). 2015. pp. 1–5.
6. Kipyatkova I., Karpov A. DNN-Based Acoustic Modeling for Russian Speech Recognition Using Kaldi. International Conference on Speech and Computer. 2016. pp. 246–253.
7. LeCun Y., Bengio Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1995. vol. 3361. no. 10. pp. 1995.
8. Abdel-Hamid O. et al. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing. 2014. vol. 22. no. 10. pp. 1533–1545.
9. Sainath T.N., Mohamed A.-r., Kingsbury B., Ramabhadran B. Deep convolutional neural networks for LVCSR. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2013. pp. 8614–8618.
10. Robinson T., Hochberg M., Renals S. The use of recurrent neural networks in continuous speech recognition. Automatic speech and speaker recognition. 1996. pp. 233–258.
11. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural computation*. 1997. vol. 9. no. 8. pp. 1735–1780.
12. Gapochkin A.V. [Neural networks in speech recognition systems]. *Science Time*. 2014. vol. 1(1). pp. 29–36. (In Russ.).
13. Graves A., Jaitly N., Mohamed A.-R. Hybrid speech recognition with deep bidirectional LSTM. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). 2013. pp. 273–278.
14. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 770–778.
15. Markovnikov N.M., Kipyatkova I., Karpov A., Filchenkov A. Deep neural networks in Russian speech recognition. Proceedings of 2017 Artificial Intelligence and Natural Language Conference. 2017. pp. 54–67.
16. Kipyatkova I., Karpov. A.A. [Variants of Deep Artificial Neural Networks for Speech Recognition Systems]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2016. vol. 49(6). pp. 80–103. (In Russ.).
17. Ackley D.H., Hinton G. E., Sejnowski T.J. A learning algorithm for Boltzmann machines. *Cognitive science*. 1985. vol. 9. no 1. pp. 147–169.
18. Srivastava N. et al. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*. 2014. vol. 15. no. 1. pp. 1929–1958.
19. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning. 2015. pp. 448–456.
20. Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*. 1996. vol. 10. pp. 707–710.
21. Mikolov T. et al. Recurrent neural network based language model. Interspeech. 2010. vol. 2. pp. 1045–1048.
22. Rao K., Peng F., Sak H., Beaufays F. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. pp. 4225–4229.
23. Jaitly N., Hinton G. Learning a better representation of speech soundwaves using restricted boltzmann machines. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2011. pp. 5884–5887.
24. Smolensky P. Information processing in dynamical systems: Foundations of harmony theory. Colorado University at Boulder Dept of Computer Science. 1986. pp. 194–281.
25. Bojarski M. et al. End to End Learning for Self-Driving Cars». 2016. preprint: arXiv: 1604.07316. Available at: https://arxiv.org/abs/1604.07316 (accessed: 17.02.2018).

26. Sayre K.M. Machine recognition of handwritten words: A project report. *Pattern recognition*. 1973. vol. 5. no. 3. pp. 213–228.
27. Graves A., Ferna´ndez S., Gomez F., Schmidhuber J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. Proceedings of the 23rd international conference on Machine learning. 2006. pp. 369–376.
28. Bridle J.S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. Neurocomputing. 1990. pp. 227–236.
29. Rabiner L.R. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE. 1989. vol. 77. no. 2. pp. 257–286.
30. Graves A., Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014. pp. 1764–1772.
31. Povey D. et al. The Kaldi speech recognition toolkit». IEEE 2011 workshop on automatic speech recognition and understanding. IEEE Signal Processing Society. 2011. 4 p.
32. Description of WSJ speech corpus. Available at: https://catalog.ldc.upenn.edu/LDC93S6B (accessed: 17.02.2018).
33. Miao Y., Gowayyed M., Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). 2015. pp. 167–174.
34. Popovi´c B., Pakoci E., Pekar D. End-to-End Large Vocabulary Speech Recognition for the Serbian Language. International Conference on Speech and Computer. 2017. pp. 343–352.
35. Mohri M., Pereira F., Riley M. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*. 2002. vol. 16. no. 1. pp. 69–88.
36. Allauzen C. et al. OpenFst: A general and efficient weighted finite-state transducer library. International Conference on Implementation and Application of Automata. 2007. pp. 11–23.
37. Collobert R., Puhrsch C., Synnaeve G. Wav2letter: an end-to-end convnetbased speech recognition system. 2016. preprint: arXiv: 1609.03193. Available at: https://arxiv.org/abs/1609.03193 (accessed: 17.02.2018).
38. Panayotov V., Chen G., Povey D., Khudanpur S. Librispeech: an ASR corpus based on public domain audio books. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. pp. 5206–5210.
39. Deep learning toolkit Torch. Available at: http://www.torch.ch/ (accessed: 17.02.2018).
40. Zhang Y. et al. Towards end-to-end speech recognition with deep convolutional neural networks. 2017. preprint: arXiv: 1701.02720. Available at: https://arxiv.org/abs/1701.02720 (accessed: 17.02.2018).
41. Description of TIMIT speech corpus. Available at: https://catalog.ldc.upenn.edu/ldc93s1 (accessed: 17.02.2018).
42. Sak H., F. de Chaumont Quitry, Sainath T., Rao K. Acoustic modelling with cd-ctc-smbr lstm rnns. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). 2015. pp. 604–609.
43. Kingsbury B. Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009). 2009. pp. 3761–3764.
44. Sainath T.N., Vinyals O., Senior A., Sak H. Convolutional, long shortterm memory, fully connected deep neural networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. pp. 4580–4584.
45. Fiscus J.G. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). Proceedings of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding. 1997. pp. 347–354.
46. Soltau H., Liao H., Sak H. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. 2016. preprint: arXiv: 1610. 09975. Available at: https://arxiv.org/abs/1610.09975. (accessed: 17.02.2018).

47. Liao H., McDermott E., Senior A. Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). 2013. pp. 368–373.
48. Youtube. Available at: https://www.youtube.com/yt/lineups/ (accessed: 17.02.2018).
49. Graves A. Sequence transduction with recurrent neural networks. 2012. preprint: arXiv: 1211.3711. Available at: https://arxiv.org/abs/1211.3711 (accessed: 17.02.2018).
50. Boulanger-Lewandowski N., Bengio Y., Vincent P. High-dimensional sequence transduction. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2013. pp. 3178–3182.
51. Graves A., Mohamed A.-R., Hinton G. Speech recognition with deep recurrent neural networks. 2013 IEEE International Conference on Acoustics, speech and signal processing (ICASSP). 2013. pp. 6645–6649.
52. Zhang Z. et al. Deep Recurrent Convolutional Neural Network: Improving Performance For Speech Recognition. 2016. preprint: arXiv: 1611.07174. Available at: https://arxiv.org/abs/1611.07174 (accessed: 17.02.2018).
53. Wang Y., Deng X., Pu S., Huang Z. Residual convolutional CTC networks for automatic speech recognition. 2017. preprint: arXiv: 1702.07793. Available at: https://arxiv.org/abs/1702.07793 (accessed: 17.02.2018).
54. Keras: The Python Deep Learning library. Available at: https://keras.io/ (accessed: 17.02.2018).
55. TensorFlow. An open source machine learning framework for everyone. Available at: https://www.tensorflow.org/ (accessed: 17.02.2018).
56. Deep learning toolkit Theano. Available at: http://deeplearning.net/software/theano/ (accessed: 17.02.2018).
57. The Microsoft Cognitive Toolkit. Available at: https://docs.microsoft.com/ru-ru/cognitive-toolkit/ (accessed: 17.02.2018).
58. Example implementation of speech recognition system. Available at: https://github.com/Microsoft/CNTK/tree/master/Tests/EndToEndTests/Speech/LSTM_CTC_MLF (accessed: 17.02.2018).
59. CTC loss-function implementation. Available at: https://github.com/baidu-research/warp-ctc (accessed: 17.02.2018).
60. CTC model implementation using Kaldi. Available at: https://github.com/lingochamp/kaldi-ctc (accessed: 17.02.2018).
61. Cho K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014. preprint: arXiv: 1406.1078. Available at: https://arxiv.org/abs/1406.1078 (accessed: 17.02.2018).
62. Sutskever I., Vinyals O., Le Q.V. Sequence to sequence learning with neural networks. Advances in neural information processing systems. 2014. pp. 3104–3112.
63. Chorowski J.K. et al. Attentionbased models for speech recognition. Advances in Neural Information Processing Systems. 2015. pp. 577–585.
64. Bahdanau D., Cho K., Bengio Y. Neural machine translation by jointly learning to align and translate. 2014. preprint: arXiv: 1409.0473. Available at: https://arxiv.org/abs/1409.0473 (accessed: 17.02.2018).
65. Mnih V., Heess N., Graves A. Recurrent models of visual attention. Advances in neural information processing systems. 2014. pp. 2204–2212.
66. Bahdanau D., et al. End-to-end attention-based large vocabulary speech recognition. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016. pp. 4945–4949.
67. Chan W., Jaitly N., Le Q., Vinyals O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016. pp. 4960–4964.
68. Kim S., Hori T., Watanabe S. Joint CTC-attention based end-to-end speech recognition using multi-task learning. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2017. pp. 4835–4839.

69. Chorowski J., Bahdanau D., Cho K., Bengio Y. End-to-end continuous speech recognition using attention-based recurrent NN: first results. 2014. preprint: arXiv: 1412.1602. Available at: https://arxiv.org/abs/1412.1602 (accessed: 17.02.2018).
70. Amodei D. et al. End to end speech recognition in English and Mandarin. ICLR 2016 workshop. 2016. 12 p.
71. Zhang Y., Chan W., Jaitly N. Very deep convolutional networks for end-to-end speech recognition. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2017. pp. 4845–4849.
72. Tjandra A., Sakti S., Nakamura S. Attention-based Wav2Text with feature transfer learning. 2017. preprint: arXiv: 1709.07814. Available at: https://arxiv.org/abs/1709.07814 (accessed: 17.02.2018).
73. Implementation of attention-based model. Available at: https://github.com/rizar/attention-lvcsr (accessed: 17.02.2018).
74. Van Merri¨enboer et al. Blocks and fuel: Frameworks for deep learning. 2015. preprint: arXiv: 1506.00619. Available at: https://arxiv.org/abs/1506.00619 (accessed: 17.02.2018).
75. Bahdanau D. et al. Task loss estimation for sequence prediction. 2015. preprint: arXiv: 1511.06456. Available at: https://arxiv.org/abs/1511.06456 (accessed: 17.02.2018).
76. Luong T., Brevdo E., Zhao R. Neural machine translation (seq2seq) tutorial. 2017. Available at: https://www.tensorflow.org/tutorials/seq2seq (accessed: 17.02.2018).
77. Implementation of end-to-end models. Available at: https://github.com/farizrahman4u/seq2seq (accessed: 17.02.2018).
78. Lafferty J., McCallum A., Pereira F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. 8 p.
79. Fosler-Lussier E., He Y., Jyothi P., Prabhavalkar R. Conditional random fields in speech, audio, and language processing. Proceedings of the IEEE. 2013. vol. 101. no. 5. pp. 1054–1075.
80. Bottou L. Une Approche th´eorique de l'Apprentissage Connexioniste; Applications `a la reconnaissance de la Parole. Ph.D. thesis. Universite de Paris XI. 1991. 236 p. (In French).
81. Hifny Y., Renals S. Speech recognition using augmented conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*. 2009. vol. 17. no. 2. pp. 354–365.
82. Kong L., Dyer C., Smith N.A. Segmental recurrent neural networks. 2015. preprint: arXiv: 1511.06018. Available at: https://arxiv.org/abs/1511.06018 (accessed: 17.02.2018).
83. Lu L., et al. Segmental recurrent neural networks for end-to-end speech recognition. 2016. preprint: arXiv: 1603.00223. Available at: https://arxiv.org/abs/1603.00223 (accessed: 17.02.2018).
84. Palaz D., Doss M.M., Collobert R. Convolutional neural networksbased continuous speech recognition using raw speech signal. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. pp. 4295–4299.
85. Amodei D., et al. Deep speech 2: End-to-end speech recognition in English and mandarin. International Conference on Machine Learning. 2016. pp. 173–182.
86. Deep learning toolkit Lasagne. Available at: http://lasagne.readthedocs.io/en/latest/ (accessed: 17.02.2018).
87. Chainer. A Powerful, Flexible, and Intuitive Framework for Neural Networks. Available at: https://chainer.org/ (accessed: 17.02.2018).
88. Dean J. et al. Large scale distributed deep networks. Advances in neural information processing systems. 2012. pp. 1223–1231.
89. Lu L., Kong L., Dyer C., Smith N.A. Multi-task Learning with CTC and Segmental CRF for Speech Recognition. 2017. preprint: arXiv: 1702.06378. Available at: https://arxiv.org/abs/1702.06378 (accessed: 17.02.2018).
90. Neubig G., et al. DyNet: The Dynamic Neural Network Toolkit. 2017. preprint: arXiv: 1701.03980. Available at: https://arxiv.org/abs/1701.03980 (accessed: 17.02.2018).