

DOI: [10.17323/2587-814X.2020.1.7.18](https://doi.org/10.17323/2587-814X.2020.1.7.18)

Нейросетевая модель для обработки запросов пользователей на этапе эксплуатации и сопровождения программного продукта

Е.И. Грибков^{a,b}

E-mail: drnemor@gmail.com

Ю.П. Ехлаков^a 

E-mail: upe@tusur.ru

^aТомский государственный университет систем управления и радиоэлектроники
Адрес: 634050, г. Томск, пр. Ленина, д. 40

^bООО «ТомскСофт»
Адрес: 634034, г. Томск, ул. Нахимова, д. 8

Аннотация

В статье предлагается оригинальная нейросетевая модель на основе системы переходов для извлечения информативных высказываний из текстов обращений пользователей. Описаны конфигурация и система переходов, определяющие процесс извлечения составных объектов как исполнение последовательности переходов. Предсказание последовательности переходов осуществляется при помощи нейронной сети, признаки которой формируются на основе конфигурации. Для обучения и оценки качества работы модели создан размеченный набор текстов отзывов о приложениях из магазина приложений Google Play для операционной системы Android. Описана процедура обучения модели для извлечения информативных фраз на наборе отзывов, предложены конкретные значения гиперпараметров модели. Проведено экспериментальное сравнение предложенной модели с гибридной моделью на основе сверточной и рекуррентной нейронных сетей. Для сравнения моделей использовался критерий F1, характеризующий соотношение полноты и точности извлечения информативных фраз. Результаты экспериментального исследования предложенной модели показывают, что она дает более высокие по сравнению с аналогом результаты: F1 для извлечения фрагментов объектов выше на 2,9%, для извлечения связей между фрагментами – на 36,2%. Качественный анализ полученных результатов говорит о возможности практического применения предложенной модели для обработки запросов пользователей при эксплуатации и сопровождении программных продуктов.

Ключевые слова: обработка естественного языка; сопровождение программного обеспечения; машинное обучение; глубокое обучение; модель на основе системы переходов.

Цитирование: Грибков Е.И., Ехлаков Ю.П. Нейросетевая модель для обработки запросов пользователей на этапе эксплуатации и сопровождения программного продукта // Бизнес-информатика. 2020. Т. 14. № 1. С. 7–18.
DOI: [10.17323/2587-814X.2020.1.7.18](https://doi.org/10.17323/2587-814X.2020.1.7.18)

Введение

Конкурентоспособность рыночного программного продукта во многом зависит от качества и своевременности реагирования ИТ-компаний на запросы конечных пользователей в связи с некачественной работой программных продуктов (ПП), ошибками в технической документации, недостаточной квалификацией конечных пользователей и т.д. Оперативное решение этих пользовательских проблем осуществляется на этапе эксплуатации и сопровождения программного продукта [1]. По данным работы [2], на эти этапы приходится 67% времени жизненного цикла ПП.

Для обработки запросов пользователей, как правило, представленных в неструктурированной текстовой форме (электронные письма, сообщения на форумах и чатах поддержки), и реагирования на них в ИТ-компаниях создаются службы технической поддержки. Следует отметить, что современные системы автоматизации работы службы поддержки (helpdesk-системы), такие как HappyFox, Service Desk Plus, FreshDesk, Zendesk, имеют достаточно развитый функционал, обеспечивающий прием и хранение заявок пользователей, связь с другими заявками, мониторинг исполнения заявок, хранение истории переписки с пользователем. Однако основная часть работы по семантическому анализу текста обращения пользователя и назначение специалистов по-прежнему возлагается на человека. При росте количества пользователей остро встает проблема масштабируемости поддержки и увеличения финансовых затрат на содержание непрофильных структур.

Кроме обработки «явных» обращений пользователей, связанных с функционалом ПП, большой интерес для компаний могут представлять «неявные» обращения, высказываемые пользователями в каналах, которые редко являются объектами внимания службы поддержки – форумах, блогах и социальных сетях за пределами созданных разработчиками ПП групп и сообществ. «Неявные» обращения, выражаемые в виде текстов мнений или отзывов, могут быть столь же полезны разработчикам ПП, как и явные, и вместе с тем являются намного более многочисленными. Поскольку ручной анализ большого объема текстовой информации затруднителен, данный пласт информации либо обрабатывается лишь время от времени, либо не рассматривается вовсе.

Задачи обработки большого объема текстовой информации могут быть решены при помощи автоматизированной обработки текстов обращений пользователей, для анализа проблем, возникших при эксплуатации ПП. В настоящее время имеется ряд работ, посвященных автоматической обработке текстов пользовательских запросов с целью выявления знаний, полезных при разработке и сопровождении программных продуктов. Так, в работе [3] для выявления жалоб на программные ошибки, содержащихся в багтрекерах проектов на основе открытого исходного кода, рассматривается применение решающих деревьев, наивного байесовского классификатора и логистической регрессии. Исследование [4] посвящено анализу различных аспектов текстов отзывов о мобильных приложениях, публикуемых в магазине Apple AppStore. Отмечается, что хотя часть отзывов не представляет интереса для разработчиков приложений, другие отзывы содержат сообщения об ошибках, пользовательский опыт и запросы на введение новых функций. Проблема анализа текстов отзывов из магазина приложений Google на предмет наличия запросов на расширение функциональности рассматривается в работе [5]. Авторы используют набор лингвистических правил для классификации предложений по критерию «содержит / не содержит запрос», после чего при помощи тематической модели Latent Dirichlet Allocation определяют основные темы текстов. В работе [6] предлагается корпус отзывов о мобильных приложениях на немецком языке из магазина приложений Google Play Store, который содержит упоминания признаков приложений (аспекты) и оценочные высказывания (описания) пользователей о них.

Проведенный анализ показывает, что существующие модели работают на уровне предложений и не позволяют непосредственно определять фразы, в которых высказывается основная суть обращения пользователя. Кроме того, каждая из описанных моделей позволяет извлекать только один тип запроса, что не соответствует реальным потребностям службы технической поддержки пользователей. В данной работе предлагается решение, способное в рамках одной модели обрабатывать множество типов пользовательских запросов и при этом выделять в текстах ключевые семантические фразы. Данная модель является развитием идей, описанных ранее в работе [7].

**1. Постановка задачи
обработки запросов пользователей
на этапе эксплуатации и сопровождения
программного продукта**

Задача обработки запросов, содержащих мнения пользователей, на этапах эксплуатации и сопровождения программного продукта в данной работе сводится к извлечению из текстов запросов информативных фраз (ИФ), содержащих конкретные вопросы, пожелания и требования пользователей. Структуру ИФ предлагается задать в виде пары «объект – описание», где под объектом понимается упоминание в тексте самого ПП, его функций, элементов графического интерфейса и т.п., а под описанием – некоторую фразу, в кото-

рой пользователь оценивает объект и описывает сложившуюся ситуацию.

Для разделения всех возможных ИФ на семантически похожие группы предложен оригинальный классификатор, основанный на элементах универсальной модели деятельности [8]: субъект деятельности, объект, на который направлена деятельность, средства, используемые в процессе деятельности, взаимосвязи между элементами деятельности. Применительно к особенностям решаемой задачи, субъектом деятельности является пользователь ПП, объектом – программный продукт, а средствами – аппаратно-программные средства, обеспечивающие функционирование ПП. На основе данной модели разработан иерархический классификатор типов ИФ, представленный на *рисунке 1*.

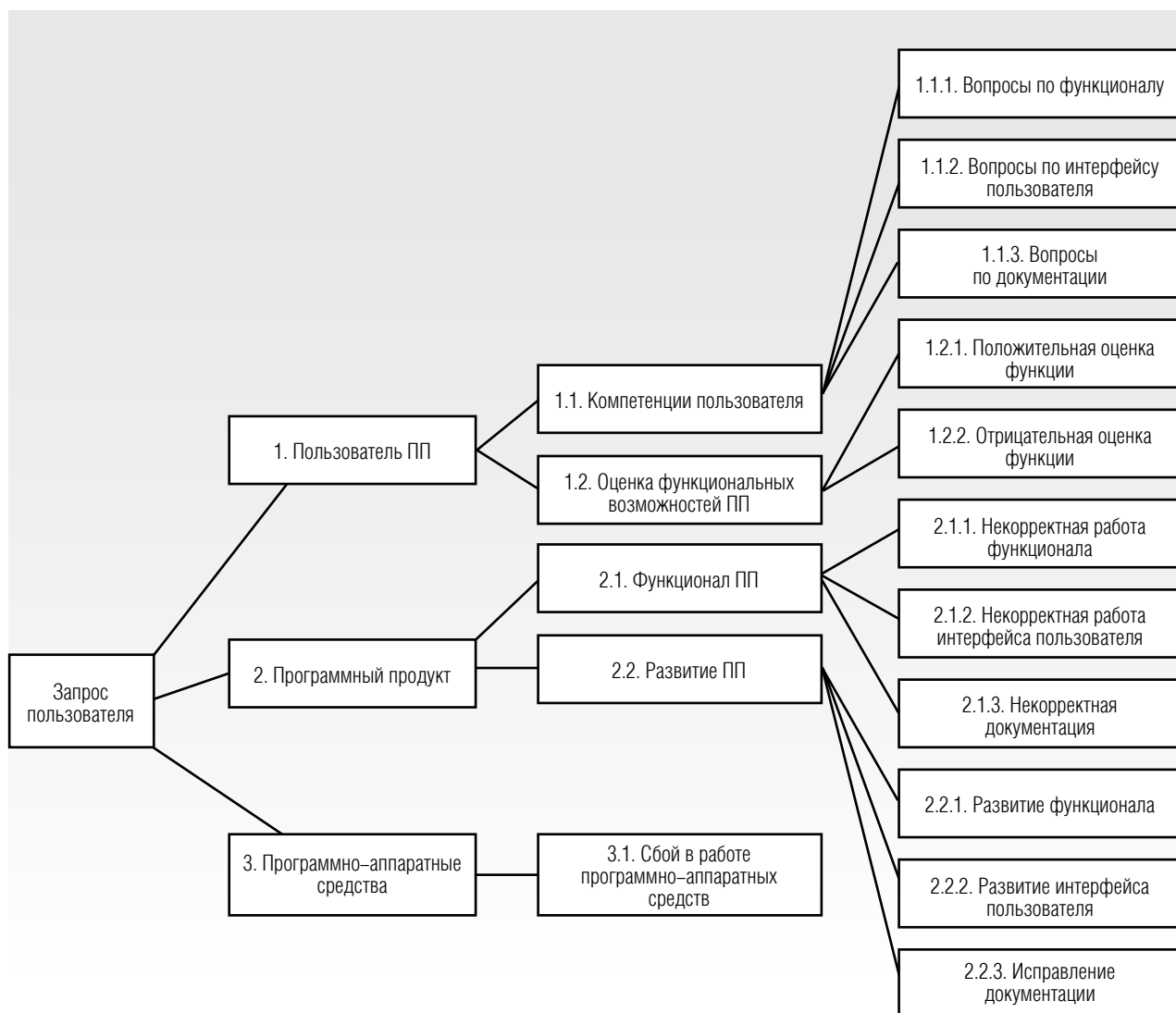


Рис. 1. Классификатор информативных фраз

К числу элементов классификатора относятся:

- ♦ *компетенции пользователя* – вопросы пользователей о функциональных возможностях ПП, его графическом интерфейсе и технической документации;

- ♦ *оценка функциональных возможностей ПП* – положительные и негативные мнения пользователей о функциональных возможностях ПП;

- ♦ *функционал ПП* – фразы, в которых пользователи сообщают о некорректной работе функций ПП и его графического интерфейса, фактических ошибках в документации;

- ♦ *развитие ПП* – запросы пользователей на развитие функциональности ПП и доработку документации и графического интерфейса;

- ♦ *сбой в работе программно-аппаратных средств* – жалобы пользователей на сбои в работе внешних по отношению к ПП программных и аппаратных средств.

Авторы не претендуют на полноту предложенного классификатора информативных фраз для любых предметных областей. В данном случае для построения модели извлечения ИФ был проведен анализ текстов запросов из магазина Google Play. В результате такого анализа из классификатора было выделено четыре укрупненных класса ИФ, чаще всего упоминаемых пользователями – программная ошибка («ошибки программного продукта»), запрос на новый функционал («развитие программного продукта»), положительная оценка функции, отрицательная оценка функции.

Приведем примеры ИФ из текстов запросов пользователей магазина приложений Google Play Market:

- ♦ *программная ошибка (bug)*: «плеер не отображается на экране блокировки», «плейлист не обновляется при свейпе вниз»;

- ♦ *запрос на новый функционал (feature request)*: «добавить кнопку проигрывания вперемешку», «иметь возможность редактировать теги»;

- ♦ *положительная оценка функции (positive)*: «много возможностей для манипуляций со звуком», «отменить прием – быстро»;

- ♦ *отрицательная оценка функции (negative)*: «не хранится архив записей», «дублирует функционал приложения госуслуги».

Таким образом, имея на входе текст запроса пользователя в виде последовательности слов $\mathbf{w} = w_1, \dots, w_N$, и структуру классификатора, необходимо извлечь из текста множество содержащихся в нем ИФ.

Формально ИФ будем называть тройку (o, d, l) , где o – объект, d – описание, l – класс ИФ из множества $Labels = \{Bug, Feature request, Positive, Negative\}$. Объект и описания в тексте определены как фрагменты (непрерывные последовательности слов): $o = (w_i, \dots, w_j)$, $d = (w_p, \dots, w_q)$. Класс ИФ будет определяться меткой из $Labels$, назначенной описанию. Примеры структуры ИФ представлен на *рисунке 2*.

2. Нейросетевая модель на основе системы переходов

Для извлечения ИФ из текстов запросов пользователей предлагается использовать оригинальную нейросетевую модель на основе системы переходов (transition-based). Данный тип моделей широко применяется при решении задач предсказания объектов со сложной структурой: определения фразовой структуры предложения [9], предсказания синтаксических деревьев зависимостей [10] и извлечения именованных сущностей [11]. Среди преимуществ моделей этого типа выделяют высокую скорость работы, а также возможность использовать нелокальные признаки. Использование модели на основе системы переходов предполагает задание абстрактного автомата, который обрабатывает входной текст и получает на выходе множество ИФ. Состояние абстрактного автомата определяется конфигурацией C , которая изменяется при исполнении автоматом переходов. Начальная конфигурация C_0 задается на основе входного текста, а после исполнения конечной последовательности переходов $y = y_1, \dots, y_T, y_i \in Y$ автомат достигает конечной конфигурации C_T , которая содержит ИФ из анализируемого текста. Выбор очередного перехода на шаге t осуществляется вероятностным классификатором на основе текущей конфигурации C_t :

$$\hat{y}_t = \operatorname{argmax}_{y \in A(C_t)} P(y | C_t). \quad (1)$$

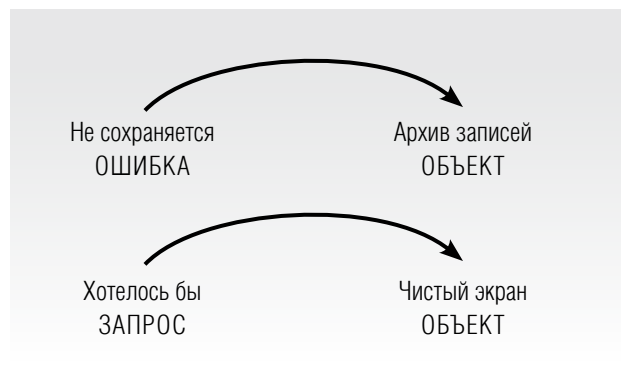


Рис. 2. Пример структуры информационной фразы

В этом случае конфигурация задана кортежем (B, S, L, H) . Список B содержит слова исходного текста. Стек S содержит объекты и описания в порядке их обнаружения в тексте. В ходе предсказания сущности из нескольких слов формируются инкрементально, поэтому в сущность на вершине стека допускается добавлять новые слова. Список L содержит связи между элементами стека S . Список H хранит историю совершенных переходов, которая привела к текущей конфигурации C_t .

Рассмотрим смысл элементов множества доступных переходов $Y = \{Shift, Start(e), Add(e), Link(n_1, n_2), End\}$. $Shift$ отбрасывает из B первый элемент. $Start(e)$ определяет новый фрагмент класса $e \in Labels \cup \{Object\}$, помещает его на вершину стека S , и перемещает слово из начала списка B в созданный фрагмент. $Add(e)$ перемещает элемент из начала списка B во фрагмент, находящийся на вершине стека S . $Link(n_1, n_2)$ связывает элементы стека S на позициях n_1 и n_2 , образованная связь помещается в конец списка L . Такой способ задания множества Y ограничивает максимальную глубину, на которой можно образовать связь между элементами S . В данной работе эту характеристику предлагается определить как наибольшее значение глубины между связанными фрагментами, встречающиеся в обучающем наборе. Переход End заканчивает процесс предсказания.

Некоторые переходы можно совершить только в том случае, если текущая конфигурация удовлетворяет некоторым условиям: например, при пустом B нельзя выполнить ни один переход, кроме End и $Link(n_1, n_2)$. Ограничения на выполнение переходов учитываются в функции $A(C_t): C \rightarrow Y' \subseteq Y$, определяющей подмножество доступных в конфигурации C_t переходов. Необходимые для совершения каждого перехода условия приведены в таблице 1.

Таблица 1.

**Предусловия
выполнения переходов**

Переход	Условие
$Shift$	$B \neq \emptyset$
$Start(e)$	$B \neq \emptyset$
$Add(e)$	$B \neq \emptyset \wedge S \neq \emptyset \wedge class(S_1) = e$
$Link(n_1, n_2)$	$\exists S_{n_1} \wedge \exists S_{n_2} \wedge (n_1, n_2) \notin L$
End	$B = \emptyset$

Условное распределение вероятности по возможным переходам из выражения (1) задается классификатором следующего вида:

$$P(y | C) = \text{softmax}(\mathbf{W} \cdot \varphi(C_t) + \mathbf{b})_y, \quad (2)$$

$$\text{softmax}(\mathbf{v}_c) = \frac{\exp(\mathbf{v}_c)}{\sum_i \exp(\mathbf{v}_i)},$$

где \mathbf{W}, \mathbf{b} – параметры классификатора;

$\varphi(C_t)$ – вектор признаков, описывающий текущую конфигурацию.

Вектор признаков (C_t) формируется путем конкатенации векторов описаний отдельных элементов конфигурации:

$$\varphi(C) = \varphi(B); \varphi(S); \varphi(H). \quad (3)$$

Основой для формирования признаков элементов конфигурации служат контекстные векторные представления для исходного текста, образуемые с помощью функции F :

$$\varphi(B_1), \dots, \varphi(B_n) = F(E(w_1), \dots, E(w_n)), \quad (4)$$

где E – функция, отображающая слова в векторные представления.

В качестве F можно использовать любые функции, позволяющие отображать одну последовательность векторов в другую. В статье рассматриваются два варианта, получившие распространение в задачах обработки естественного языка: одномерные сверточные (CNN) [12–14] и рекуррентные (RNN) [15–17] нейронные сети.

В случае CNN каждый вектор выходной последовательности на позиции n формируется на основе векторов входной последовательности, находящихся в окне размера k с центром n по следующей формуле:

$$\mathbf{h}_n = \sigma(\mathbf{W}(E(w_{n-(k-1)/2}); \dots; E(w_{n+(k-1)/2})) + \mathbf{b}), \quad (5)$$

где \mathbf{W}, \mathbf{b} – параметры свертки;

σ – нелинейная функция активации.

Таким образом, \mathbf{h}_n можно назвать контекстным представлением исходного слова на позиции n . В качестве σ предлагается использовать функцию активации MaxOut [18], так как она показала наилучшую точность в ходе предварительных экспериментов. Кроме того, для улучшения качества обучения сети был использован подход из работы [19], суть которого заключается в том, что выход слоя сумми-

руется с изначальным входом. Таким образом, выход слоя описывается выражением:

$$\mathbf{h}_n^i = \sigma(\mathbf{W}\mathbf{h}_{n-1}^i + \mathbf{b}), + \mathbf{h}_n^{i-1}. \quad (6)$$

При использовании рекуррентных нейронных сетей (РНС) контекстные представления элементов последовательности образуются с помощью рекуррентных взаимосвязей следующего вида:

$$\mathbf{h}_n = \sigma(\mathbf{U}E(w_n) + \mathbf{W}\mathbf{h}_{n-1} + \mathbf{b}). \quad (7)$$

В этом случае контекстная информация передается таким образом, что элемент n несет информацию обо всех предыдущих $(n - 1)$. Учитывать последующий контекст можно применив двунаправленные РНС [20], в которых итоговые контекстные векторы образуются путем объединения векторов $\bar{\mathbf{h}}_n$ и $\tilde{\mathbf{h}}_n$, полученные при прямом (слева направо) и обратном (справа налево) проходах:

$$\begin{aligned} \bar{\mathbf{h}}_n &= \sigma(\bar{\mathbf{U}}E(w_n) + \bar{\mathbf{W}}\bar{\mathbf{h}}_{n-1} + \bar{\mathbf{b}}), \\ \tilde{\mathbf{h}}_n &= \sigma(\tilde{\mathbf{U}}E(w_n) + \tilde{\mathbf{W}}\tilde{\mathbf{h}}_{n+1} + \tilde{\mathbf{b}}), \\ \mathbf{h}_n &= (\bar{\mathbf{h}}_n; \tilde{\mathbf{h}}_n). \end{aligned} \quad (8)$$

Наиболее распространенным видом РНС является LSTM [21], показывающий высокую точность при решении многих задач обработки естественного языка [22–24]. В нашей работе предлагается использовать двунаправленную LSTM-сеть (Bi-LSTM) с несколькими слоями. На рисунке 3 приведены диаграммы сверточной и рекуррентной сетей для формирования контекстных представлений.

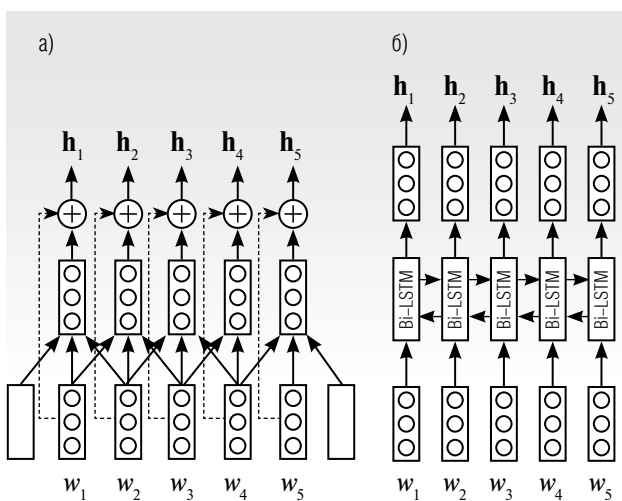


Рис. 3. Получение контекстных представлений элементов последовательности с помощью CNN и LSTM: сверточная нейронная сеть (а), рекуррентная нейронная сеть Bi-LSTM (б)

Вектор $\varphi(B)$ образован конкатенацией признаков первых трех элементов списка B :

$$\varphi(B) = [\varphi(B_1); \varphi(B_2); \varphi(B_3)].$$

Для вектора $\varphi(S)$ сначала рассчитываются векторы признаков для отдельных элементов S по следующей формуле:

$$\varphi(S_i) = \left(\frac{1}{e(i) - b(i)} \sum_{j=e(i)}^{b(i)} \varphi(B_j); E(\text{type}(i)) \right), \quad (9)$$

где $b(i)$, $e(i)$ – начальный и конечный индексы i -го фрагмента текста;

$E(\text{type}(i))$ – векторное представление типа фрагмента.

Поскольку в обучающей выборке максимальное расстояние между двумя любыми объединенными связью фрагментами равно четырем, глубину поиска связи можно ограничить первыми пятью элементами S . В этом случае полученные признаки для первых пяти элементов S конкатенируются, образуя вектор $\varphi(S)$:

$$\varphi(S) = (\varphi(S_1); \dots; \varphi(S_5)). \quad (10)$$

Для задания признаков истории переходов H в момент времени t используется скрытое состояние последнего шага сети LSTM:

$$\varphi(H) = \text{LSTM}(E(H_1); \dots; E(H_t))_t. \quad (11)$$

Предложенная модель (1–11) позволяет использовать одни и те же компоненты для получения признаков, которые используются как при извлечении фрагментов, так и при определении связей между ними. Это уменьшает общее количество параметров модели, которые необходимо обучить, что препятствует переобучению. Кроме того, такая структура модели реализует принципы многозадачного обучения [25, 26] (multitask learning), когда использование одной модели для решения нескольких связанных задач ускоряет процесс обучения и/или улучшает показатели качества. Обучение модели осуществляется путем максимизации правдоподобия истинных последовательностей переходов, которые формируются из предложений обучающей выборки. Функцией ошибки служит перекрестная энтропия (cross entropy) между предсказанным распределением по возможным действиям и истинным действиям:

$$L(\hat{\mathbf{y}}, \mathbf{y}) = \sum_t \left(- \sum_{i \in Y} \log(P(\hat{y}_t | C_t))_i(\mathbf{y}_t)_i \right). \quad (12)$$

Для оптимизации множества параметров модели θ , включающего параметры классификатора, сетей BiLSTM и LSTM и векторных представлений $E(H)$ и $E(type(i))$, может использоваться любой алгоритм градиентной оптимизации. Итоговая архитектура нейронной сети приведена на рисунке 4.

3. Экспериментальное исследование модели и анализ результатов

Валидация нейросетевой модели была проведена на корпусе текстов запросов пользователей магазина приложений Google Play на русском языке, состоящем из девяти категорий мобильных приложений: «Автомобили и транспорт», «Карты и навигация», «Медицина», «Музыка и аудио», «Персонализация», «Финансы», «Шопинг», «Образование», «Видеоплееры и редакторы». В каждой категории было выбрано по пять приложений, для каждого из них случайным образом было выбрано по 20 обращений. Каждое обращение, в свою очередь, разделилось на отдельные предложения.

Разметка запросов пользователей осуществлялась авторами работы на протяжении трех недель, после чего результат разметки был представлен стороннему специалисту для аудита и поиска ошибок. Затем

на основе полученной обратной связи в разметку были внесены исправления. Количественные данные по собранному корпусу приведены в таблице 2.

Таблица 2.

Количественные данные по корпусу Google Play Store

Количество текстов запросов	900
Среднее количество слов в запросе	52,3
Количество ИФ	2391
Среднее количество ИФ в запросе	2,65
Количество объектов	2273
Количество положительных оценок функций	999
Количество негативных оценок функций	851
Количество запросов на новый функционал	200
Количество программных ошибок	677

В ходе экспериментов были опробованы два варианта предложенной модели, использующие в качестве механизма получения контекстных представлений сверточную нейронную сеть (6) и двунаправленную LSTM (8), которые далее будут называться Trans-CNN и Trans-LSTM. Для Trans-CNN были заданы следующие значения параметров:

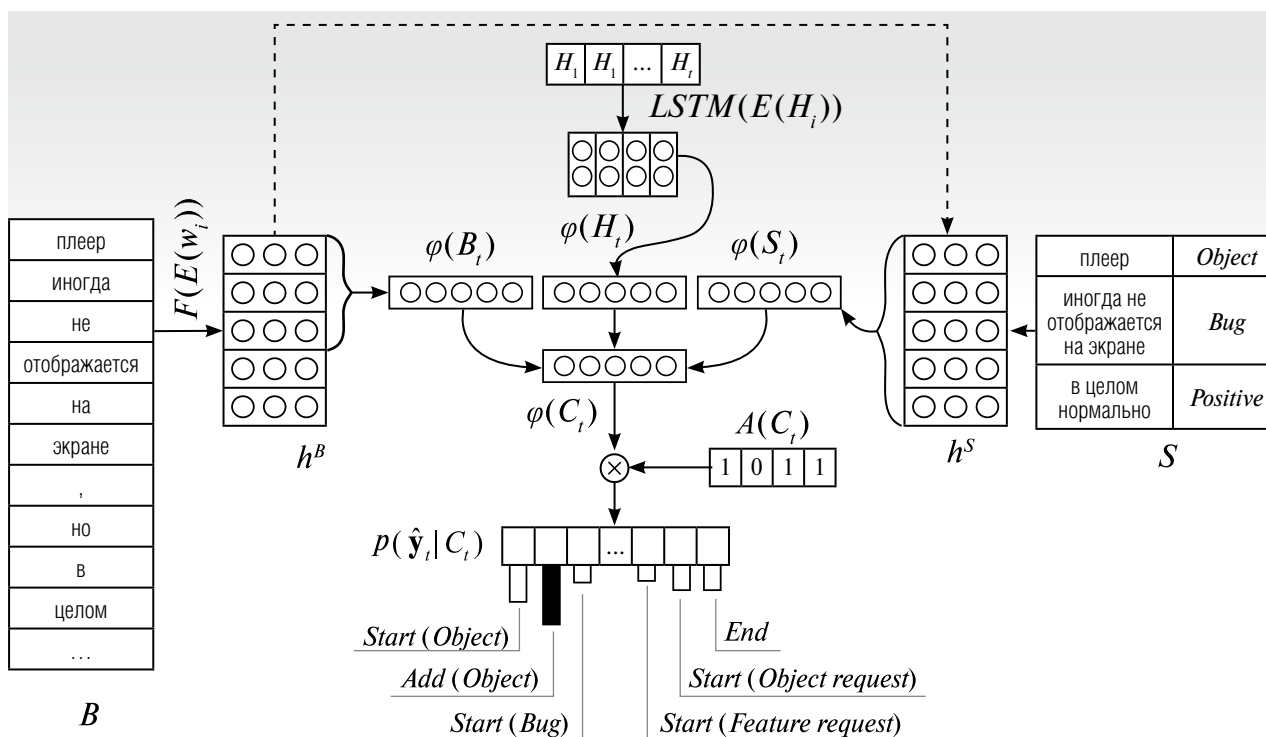


Рис. 4. Архитектура нейронной сети

размер окна свертки – 3, количество сверточных слоев – 3, количество фильтров – 150. В случае Trans-LSTM количество слоев BiLSTM – 2, размер скрытого слоя BiLSTM – 200. Общие параметры модели: размер скрытого слоя LSTM_н – 30, представление $E(H_i)$ – 30, $E(class(i))$ – 30.

Векторные представления слов строятся на основе предварительно обученной модели fastText [27] для русского языка, размер векторных представлений – 300. Выбор в пользу fastText обусловлен устойчивостью данной модели к опечаткам, а также эмпирическими данными, свидетельствующими о том, что он обладает лучшей точностью при работе с языками с богатой морфологией (например, русским). Оптимизация параметров модели осуществляется методом Adam со скоростью обучения 10^{-3} . Для предотвращения переобучения используются техники регуляризации: дропаут (dropout) 20%, регуляризация L_2 нормы весов на уровне $1,2 \cdot 10^{-6}$. В качестве базовой альтернативы для сравнения использовалась гибридная модель на основе сверточной и рекуррентной сети, предложенная в работе [28].

Вычисление оценок качества извлечения проводилось с помощью процедуры k -fold кросс-валидации, при которой выборка разбивается на k непересекающихся частей одинакового размера, после чего производится k итераций обучения модели на $k-1$ частях и тестирование на части, которая не использовалась при обучении. Каждая часть выборки используется в качестве тестовой один раз. Набор из k полученных оценок усредняется. В

нашем случае части формировались из обращений, относящихся к одной и той же категории приложений. Соответственно, в итоге для обучения и проверки использовалось девять частей. Результатом такого способа получения оценок качества является «пессимистическая» оценка: во время обучения модель не имеет доступа к лексике, специфичной для определенной категории, что повышает требования к обобщающей способности модели.

Результаты сравнения двух вариантов модели на основе системы переходов и гибридной модели приведены в *таблице 3*. Значение F-меры для фрагментов считалось усреднением значений F-меры, полученной для каждого из пяти типов фрагментов. Trans-LSTM обладает наилучшей точностью как при извлечении фрагментов ИФ – 67% F1, так и при определении связей между фрагментами – 64,8%. Улучшение качества извлечения относительно базовой модели составляет 2,9% для извлечения фрагментов и 36,2% для определения связей.

Таблица 3.

Результаты сравнения качества моделей

Модель	Фрагменты F1	Связи F1
Базовая	0,651	0,476
Trans-CNN	0,641	0,598
Trans-LSTM	0,670	0,648

Детальное сравнение результатов обучения Trans-LSTM и базовой модели в разрезе каждой катего-

Таблица 4.

Сравнение Trans-LSTM с базовой моделью

Категория	Объект	Положительная оценка	Отрицательная оценка	Ошибка	Запрос на новый функционал	Связь
Авто и транспорт	0,747 (0,730)	0,724 (0,639)	0,548 (0,672)	0,646 (0,678)	0,691 (0,633)	0,676 (0,493)
Персонализация	0,721 (0,684)	0,685 (0,677)	0,602 (0,559)	0,598 (0,625)	0,602 (0,658)	0,633 (0,459)
Музыка и аудио	0,748 (0,667)	0,649 (0,649)	0,624 (0,535)	0,630 (0,594)	0,632 (0,542)	0,603 (0,431)
Карты и навигация	0,656 (0,699)	0,628 (0,665)	0,559 (0,529)	0,711 (0,679)	0,694 (0,481)	0,631 (0,455)
Медицина	0,707 (0,711)	0,713 (0,600)	0,713 (0,637)	0,691 (0,617)	0,673 (0,591)	0,659 (0,490)
Финансы	0,708 (0,706)	0,659 (0,709)	0,615 (0,626)	0,641 (0,623)	0,622 (0,700)	0,682 (0,497)
Шопинг	0,745 (0,715)	0,788 (0,756)	0,697 (0,647)	0,576 (0,685)	0,841 (0,691)	0,689 (0,524)
Образование	0,746 (0,762)	0,695 (0,745)	0,586 (0,623)	0,757 (0,726)	0,812 (0,702)	0,686 (0,511)
Видеоплееры и редакторы	0,677 (0,693)	0,707 (0,685)	0,534 (0,647)	0,608 (0,531)	0,529 (0,565)	0,574 (0,422)
Среднее	0,717 (0,707)	0,694 (0,681)	0,609 (0,608)	0,651 (0,640)	0,677 (0,618)	0,648 (0,476)

рии приведено в *таблице 4*, где число в скобках означает точность альтернативной модели, а жирным выделен лучший результат.

Результаты апробации позволяют говорить о том, что модель показала хорошие результаты при извлечении информативных фраз и может быть использована для решения практических задач.

Заключение

Результаты проведенных экспериментов позволяют говорить о том, что предложенная модель по сравнению с базовой показывает лучшие результаты при решении задачи обработки запросов пользователей. Улучшение относительно базовой модели составляет 2,9% для извлечения фрагментов и 36,2% для определения связей. Качественный анализ полученных результатов говорит о возможности практического применения предложенной модели для обработки запросов пользователей при эксплуатации и сопровождении программных продуктов.

Извлеченная в соответствии с предложенным классификатором информация из запросов пользователей может быть использована:

- ◆ службой поддержки для распределения запросов пользователей, сгруппированных в соответ-

ствии основаниями классификации по специалистам с необходимыми компетенциями, выделения стандартных запросов и автоматического формирования ответов на базе типовых шаблонов;

- ◆ отделом разработки ПП для исправления массовых критических ошибок на основе анализа претензий и предложений пользователей, модификации функциональных и нефункциональных требований, улучшения дизайна, корректировки документации пользователей;

- ◆ отделом маркетинга для постоянного мониторинга на основе анализа негативных и позитивных оценок качества как собственных продуктов, так и продуктов конкурентов, и модификации комплекса маркетинговых мероприятий на основе этой информации.

Благодарности

Работа выполнена в рамках государственного задания Министерства науки и образования РФ, проект «Методическое и инструментальное обеспечение принятия решений в задачах управления социально-экономическими системами и процессами в гетерогенной информационной среде». ■

Литература

1. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. М.: Стандартинформ, 2011.
2. Schach S.R. Object-oriented and classical software engineering. N.Y.: McGraw-Hill Education, 2011.
3. Is it a bug or an enhancement? A text-based approach to classify change requests / G. Antoniol [et al.] // Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds. Ontario, Canada, 27–30 October 2008. P. 304–318. DOI: 10.1145/1463788.1463819.
4. Pagano D., Maalej W. User feedback in the appstore: An empirical study // Proceedings of the 21st IEEE International Requirements Engineering Conference. Rio de Janeiro, Brasil, 15–19 July 2013. P. 125–134. DOI: 10.1109/RE.2013.6636712.
5. Iacob C., Harrison R. Retrieving and analyzing mobile apps feature requests from online reviews // Proceedings of the 10th Working Conference on Mining Software Repositories (MSR 2013). San Francisco, USA, 18–19 May 2013. P. 41–44. DOI: 10.1109/MSR.2013.6624001.
6. SCARE – The sentiment corpus of app reviews with fine-grained annotations in German / M. Sängler [et al.] // Proceedings of the 10th International Conference on Language Resources and Evaluation. Portorož, Slovenia, 23–28 May 2016. P. 1114–1121.
7. Ехлаков Ю.П., Грибков Е.И. Модель извлечения пользовательских мнений о потребительских свойствах товара на основе рекуррентной нейронной сети // Бизнес-информатика. 2018. № 4 (46). С. 7–16. DOI: 10.17323/1998-0663.2018.4.7.16.
8. Перегудов Ф.И., Тарасенко Ф.П. Основы системного анализа. Томск: НТЛ, 1997.
9. Dyer C., Kuncoro A., Ballesteros M., Smith N.A. Recurrent neural network grammars // Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, USA, 12–17 June 2016. P. 199–209. DOI: 10.18653/v1/N16-1024.
10. Kiperwasser E., Goldberg Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations // Transactions of the Association for Computational Linguistics. 2016. Vol. 4. P. 313–327. DOI: 10.1162/tacl_a_00101.
11. Neural architectures for named entity recognition / G. Lample [et al.] // Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, USA, 12–17 June 2016. P. 260–270. DOI: 10.18653/v1/N16-1030.

12. Kim Y. Convolutional neural networks for sentence classification // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar, 25–29 October 2014. P. 1746–1751. DOI: 10.3115/v1/D14-1181.
13. Convolutional sequence to sequence learning / J. Gehring [et al.] // Proceedings of the 34th International Conference on Machine Learning (ICML). Sydney, Australia, 6–11 August 2017. Vol. 70. P. 1243–1252.
14. Kalchbrenner N., Grefenstette E., Blunsom P. A convolutional neural network for modelling sentences // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, USA, 22–27 June 2014. P. 655–665. DOI: 10.3115/v1/P14-1062.
15. Huang Z., Xu W., Yu K. Bidirectional LSTM-CRF models for sequence tagging // arXiv.org. [Электронный ресурс]: <https://arxiv.org/abs/1508.01991> (дата обращения 20.01.2020).
16. Ārsoy O., Cardie C. Opinion mining with deep recurrent neural networks // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar, 25–29 October 2014. P. 720–728. DOI: 10.3115/v1/D14-1080.
17. Wang W., Jialin Pan S., Dahlmeier D., Xiao X. Recursive neural conditional random fields for aspect-based sentiment analysis // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP). Austin, USA, 1–5 November 2016. P. 616–626. DOI: 10.18653/v1/D16-1059.
18. Maxout networks / I. Goodfellow [et al.] // Proceedings of the 30th International Conference on Machine Learning. Atlanta, USA, 16–21 June 2013. P. 1319–1327.
19. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition // Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las-Vegas, USA, 26 June – 1 July 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90.
20. Graves A., Jaitly N., Mohamed A. Hybrid speech recognition with deep bidirectional LSTM // Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding. Olomouc, Czech Republic, 8–12 December 2013. P. 273–278. DOI: 10.1109/ASRU.2013.6707742.
21. Hochreiter S., Schmidhuber J. Long short-term memory // Neural Computation. 1997. Vol. 9. No 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
22. Chiu J.P.C., Nichols E. Named entity recognition with bidirectional LSTM-CNNs // Transactions of the Association for Computational Linguistics. 2016. No 4. P. 357–370. DOI: 10.1162/tacl_a_00104.
23. Ma X., Hovy E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany, 7–12 August, 2016. Vol. 1. P. 1064–1074. DOI: 10.18653/v1/P16-1101.
24. Wang Y., Huang M., Zhu X., Zhao L. Attention-based LSTM for aspect-level sentiment classification // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP). Austin, USA, 1–5 November 2016. P. 606–615. DOI: 10.18653/v1/D16-1058.
25. Caruana R. Multitask learning: A knowledge-based source of inductive bias // Proceedings of the Tenth International Conference on Machine Learning. Amherst, USA, 27–29 June 1993. P. 41–48. DOI: 10.1016/b978-1-55860-307-3.50012-5.
26. Hashimoto K., Xiong C., Tsuruoka Y., Socher R. A joint many-task model: Growing a neural network for multiple NLP tasks // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP). Copenhagen, Denmark, 7–11 September 2017. P. 1923–1933. DOI: 10.18653/v1/D17-1206.
27. Learning word vectors for 157 languages / E. Grave [et al.] // Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018). Miyazaki, Japan, 7–12 May 2018. P. 3483–3487.
28. Jebbara S., Cimiano P. Aspect-based relational sentiment analysis using a stacked neural network architecture // Proceedings of the 22nd European Conference on Artificial Intelligence. The Hague, The Netherlands, 29 August – 2 September 2016. P. 1123–1131. DOI: 10.3233/978-1-61499-672-9-1123.

Об авторах

Грибков Егор Игоревич

аспирант, кафедра автоматизации обработки информации,
Томский государственный университет систем управления и радиоэлектроники,
634050, г. Томск, пр. Ленина, д. 40;
специалист по машинному обучению, ООО «ТомскСофт»,
634034, г. Томск, ул. Нахимова, д. 8;
E-mail: drnemor@gmail.com

Ехлаков Юрий Поликарпович

доктор технических наук, профессор;
профессор, кафедра автоматизации обработки информации,
Томский государственный университет систем управления и радиоэлектроники,
634050, г. Томск, пр. Ленина, д. 40;
E-mail: ure@tusur.ru
ORCID: 0000-0003-1662-4005

Neural network model for user request analysis during software operations and maintenance phase

Egor I. Gribkov^{a,b}

E-mail: drnemor@gmail.com

Yuri P. Yekhlakov^a

E-mail: upe@tusur.ru

^a Tomsk State University of Control Systems and Radioelectronics
Address: 40, Prospect Lenina, Tomsk 634050

^b TomskSoft LLC
Address: 8, Nahimova Street, Tomsk 634034

Abstract

This article offers a transition-based neural network model for extracting informative expressions from user request texts. The configuration and transition system that turns the process of informative expression extraction into the execution of a sequence of transitions is described. Prediction of transition sequence is done using a neural network that uses features derived from the configuration. To train and evaluate a proposed model, a corpus of annotated Android mobile application reviews from the Google Play store was created. The training procedure of the model for informative expressions extraction and selected model's hyperparameters are described. An experiment was conducted comparing the proposed model and an alternative model based on a hybrid of convolutional and recurrent neural networks. To compare quality of these two models, the F1 score that aggregates recall and precision of extracted informative expressions was used. The experiment shows that the proposed model extracts expressions of interest better than the alternative: the F1 score for spans extraction increased by 2.9% and the F1 for link extraction increased by 36.2%. A qualitative analysis of extracted expressions indicates that the proposed model is applicable for the task of user request analysis during operation and the maintenance phase of software products.

Key words: natural language processing; software maintenance; machine learning; deep learning; transition-based model.

Citation: Gribkov E.I., Yekhlakov Yu.P. (2020) Neural network model for user request analysis during software operations and maintenance phase. *Business Informatics*, vol. 14, no 1, pp. 7–18. DOI: 10.17323/2587-814X.2020.1.7.18

References

1. Standartinform (2011) *GOST R ISO/IEC 12207-2010. Information technology. Systems and software engineering. Software life cycle processes*. Moscow: Standartinform (in Russian).
2. Schach S.R. (2011) *Object-oriented and classical software engineering*. N.Y.: McGraw-Hill Education.
3. Antoniol G., Ayari K., Di Penta M., Khomh F., Guéhéneuc Y.-G. (2008) Is it a bug or an enhancement? A text-based approach to classify change requests. Proceedings of the *2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds, Ontario, Canada, 27–30 October, 2008*, pp. 304–318. DOI: 10.1145/1463788.1463819.
4. Pagano D., Maalej W. (2013) User feedback in the appstore: An empirical study. Proceedings of the *21st IEEE International Requirements Engineering Conference. Rio de Janeiro, Brasil, 15–19 July 2013*, pp. 125–134. DOI: 10.1109/RE.2013.6636712.
5. Iacob C., Harrison R. (2013) Retrieving and analyzing mobile apps feature requests from online reviews. Proceedings of the *10th Working Conference on Mining Software Repositories (MSR 2013), San Francisco, USA, 18–19 May 2013*, pp. 41–44. DOI: 10.1109/MSR.2013.6624001.
6. Sängler M., Leser U., Kemmerer S., Adolphs P., Klinger R. (2016) SCARE – The sentiment corpus of app reviews with fine-grained annotations in German. Proceedings of the *10th International Conference on Language Resources and Evaluation. Portorož, Slovenia, 23–28 May 2016*, pp. 1114–1121.
7. Yekhlakov Yu.P., Gribkov E.I. (2018) User opinion extraction model concerning consumer properties of products based on a recurrent neural network. *Business Informatics*, vol. 46, no 4, pp. 7–16. DOI: 10.17323/1998-0663.2018.4.7.16.
8. Peregudov F.I., Tarasenko F.P. (1997) *Basics of system analysis: guide*. Tomsk: NTL (in Russian).
9. Dyer C., Kuncoro A., Ballesteros M., Smith N.A. (2016) Recurrent neural network grammars. Proceedings of the *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, USA, 12–17 June 2016*, pp. 199–209. DOI: 10.18653/v1/N16-1024.

10. Kiperwasser E., Goldberg Y. (2016) Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 313–327. DOI: 10.1162/tacl_a_00101.
11. Lample G., Ballesteros M., Subramanian S., Kawakami K., Dyer C. (2016) Neural architectures for named entity recognition. Proceedings of the *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, USA, 12–17 June 2016*, pp. 260–270. DOI: 10.18653/v1/N16-1030.
12. Kim Y. (2014) Convolutional neural networks for sentence classification. Proceedings of the *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014*, pp. 1746–1751. DOI: 10.3115/v1/D14-1181.
13. Gehring J., Auli M., Grangier D., Yarats D., Dauphin Y.N. (2017) Convolutional sequence to sequence learning. *Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017*, vol. 70, pp. 1243–1252.
14. Kalchbrenner N., Grefenstette E., Blunsom P. (2014) A convolutional neural network for modelling sentences. Proceedings of the *52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, USA, 22–27 June 2014*, vol. 1, pp. 655–665. DOI: 10.3115/v1/P14-1062.
15. Huang Z., Xu W., Yu K. (2015) Bidirectional LSTM-CRF models for sequence tagging. *arXiv.org*. Available at: <https://arxiv.org/abs/1508.01991> (accessed 20 January 2020).
16. İrsoy O., Cardie C. (2014) Opinion mining with deep recurrent neural networks. Proceedings of the *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014*, pp. 720–728. DOI: 10.3115/v1/D14-1080.
17. Wang W., Jialin Pan S., Dahlmeier D., Xiao X. (2016) Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis. Proceedings of the *2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), Austin, USA, 1–5 November 2016*, pp. 616–626. DOI: 10.18653/v1/D16-1059.
18. Goodfellow I., Warde-Farley D., Mirza M., Courville A., Bengio Y. (2013) Maxout networks. Proceedings of the *30th International Conference on Machine Learning, Atlanta, USA, 16–21 June 2013*, pp. 1319–1327.
19. He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. Proceedings of the *2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 26 June – 1 July 2016*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
20. Graves A., Jaitly N., Mohamed A. (2013) Hybrid speech recognition with deep bidirectional LSTM. Proceedings of the *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December, 2013*, pp. 273–278. DOI: 10.1109/ASRU.2013.6707742.
21. Hochreiter S., Schmidhuber J. (1997) Long short-term memory. *Neural Computation*, vol. 9, no 8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
22. Chiu J. P.C., Nichols E. (2016) Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, no 4, pp. 357–370. DOI: 10.1162/tacl_a_00104.
23. Ma X., Hovy E. (2016) End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. Proceedings of the *54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016*, pp. 1064–1074. DOI: 10.18653/v1/P16-1101.
24. Wang Y., Huang M., Zhu X., Zhao L. (2016) Attention-based LSTM for aspect-level sentiment classification. Proceedings of the *2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), Austin, USA, 1–5 November 2016*, pp. 606–615. DOI: 10.18653/v1/D16-1058.
25. Caruana R. (1993) Multitask learning: A knowledge-based source of inductive bias. Proceedings of the *10th International Conference on International Conference on Machine Learning, Amherst, USA, 27–29 June 1993*, pp. 41–48. DOI: 10.1016/b978-1-55860-307-3.50012-5.
26. Hashimoto K., Xiong C., Tsuruoka Y., Socher R. (2017) A joint many-task model: Growing a neural network for multiple NLP tasks. Proceedings of the *2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), Copenhagen, Denmark, 7–11 September 2017*, pp. 1923–1933. DOI: 10.18653/v1/D17-1206.
27. Grave E., Bojanowski P., Gupta P., Joulin A., Mikolov T. (2018) Learning word vectors for 157 languages. *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7– 12 May 2018*, pp. 3483–3487.
28. Jebbara S., Cimiano P. (2016) Aspect-based relational sentiment analysis using a stacked neural network architecture. Proceedings of the *22nd European Conference on Artificial Intelligence, The Hague, The Netherlands, 29 August – 2 September, 2016*, pp. 1123–1131. DOI: 10.3233/978-1-61499-672-9-1123.

About the authors

Egor I. Gribkov

Doctoral Student, Department of Data Processing Automation, Tomsk State University of Control Systems and Radioelectronics, 40, Prospect Lenina, Tomsk 634050, Russia;

Machine Learning Engineer, TomskSoft LLC, 8, Nahimova Street, Tomsk 634034, Russia;

E-mail: drmemor@gmail.com

Yuri P. Yekhlakov

Dr. Sci. (Tech.), Professor;

Professor, Department of Data Processing Automation, Tomsk State University of Control Systems and Radioelectronics, 40, Prospect Lenina, Tomsk 634050, Russia;

E-mail: upe@tusur.ru

ORCID: 0000-0003-1662-4005