УДК 519.766

И.Д. Чернобаев, А.С. Суркова, А.З. Панкратова

МОДЕЛИРОВАНИЕ ТЕКСТОВ С ИСПОЛЬЗОВАНИЕМ РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ

Нижегородский государственный технический университет им. Р. Е. Алексеева

Посвящена построению и исследованию модели текстов на естественном языке, основанных на использовании рекуррентных нейронных сетей. Авторами предложена модификация модели за счет применения ненасыщаемой логарифмической функции активации.

Рассмотрено использование предложенной методики построения модели для задач автоматического реферирования текстов, сравнение со стандартными методами показало лучшие результаты для предложенной модификации.

Ключевые слова: нейронная сеть, рекуррентная нейронная сеть, Long-Short-Term-Memory, функция активации, обработка естественных языков, Encoder-Decoder.

Введение

Моделирование текстов является актуальной задачей в связи с непрерывным ростом информации, доступной в текстовом виде. Данная задача является подзадачей в таких областях, как анализ тональности текста, автоматическое реферирование, определение автора текста, классификация и кластеризация текстовых данных.

Существует множество различных видов моделей, наиболее распространенными из которых являются: векторная модель, модель на основе N-грамм, и модели на основе нейронных сетей, которые в свою очередь подразделяются на Word2Vec, CBOW, skip-gram и другие [1]. Векторная модель рассматривает текст как неупорядоченное множество слов и применяется при решении задач классификации и кластеризации документов, реферирования текстов, поиска документов по запросу. Модель на основе N-грамм основана на применении статистического подхода и используется в задачах, связанных с распознаванием речи, машинным обучением и сжатием данных. Нейросетевые модели успешно применяются при решении задачи машинного перевода, аннотирования изображений, распознавания речи, классификации текстов благодаря способности нейронных сетей автоматически усваивать закономерности обучающей выборки. Актуальность нейросетевых моделей обусловлена тем, что при правильном обучении такие модели превосходят альтернативные.

Целью данной работы является исследование модели представлений текстов на основе рекуррентной нейронной сети.

Теоретический анализ

На сегодняшний день моделирование текстов с использованием нейронных сетей является одним из наиболее успешных методов языкового моделирования. Языковое моделирование ставит своей задачей выявить закономерности естественного языка, важные при решении конкретных задач.

Языковая модель [2] позволяет определить вероятность предложения как:

$$P(w_1, w_2, ..., w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, ..., w_{i-1}),$$

где w_i – слова предложения; $P(w_1, w_2, ..., w_m)$ – вероятность предложения; $P(w_i \mid w_1, ..., w_{i-1})$ – вероятность появление слова w_i после последовательности слов $w_1, ..., w_{i-1}$; m – число слов в предложении.

Вероятность предложения определяется произведением условных вероятностей каждого слова при условии наличия предшествующих слов. Например, вероятность предложения "Рукописи не горят" определяется как вероятность слова "горят" при условии "Рукописи не", умноженная на вероятность "не" при условии "Рукописи":

$$P(\text{"Рукописи не горят"}) = P(\text{"горят"}|\text{"Рукописи не"}) * P(\text{"не"}|\text{"Рукописи"})$$

Способность учета предшествующих элементов последовательности – отличительная особенность рекуррентной нейронной сети (РНС) [3]. Нелинейная функция активации используется повсеместно в нейронных сетях для осуществления процесса обучения.

При обучении стандартных РНС на длинных предложениях возникает проблема исчезающего градиента. Суть ее заключается в том, что значение производной от нелинейной функции активации может принимать значения, близкие к нулю (рис. 1), которые, в свою очередь, передаются через цепь произведений и приводят к «исчезновению» итоговых изменений градиента.

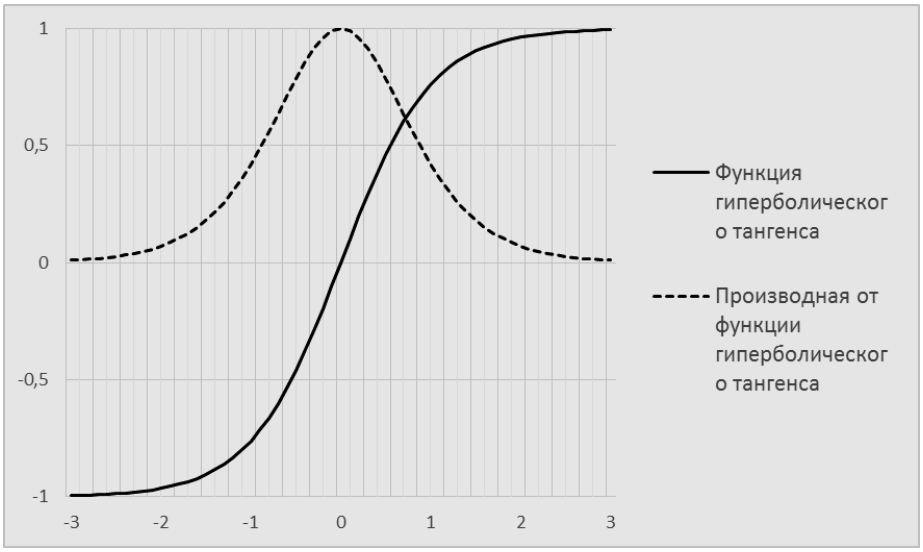

Рис. 1. График функции гиперболического тангенса и его производной

Для решения данной проблемы была разработана специальная архитектура РНС - сеть с долговременной-краткосрочной памятью (Long-Short-Term-Memory (LSTM)), представленная в [4]. Данный тип сетей широко применяется в области обработки естественного языка. Противостоять проблеме исчезающего градиента LSTM сети позволяет механизм фильтров. Этот механизм дает возможность регулировать поступление новой информации в вектор состояния c_t сети, а также вывод состояния h_t сети и обновление ее состояния c_t. Векторы фильтров сети определяются по формулам:

$$i_t = \sigma(U_i \cdot x_t + W_i \cdot h_{t-1}), \quad (1)$$

$$f_t = \sigma(U_f \cdot x_t + W_f \cdot h_{t-1}), \quad (2)$$

$$o_t = \sigma(U_o \cdot x_t + W_o \cdot h_{t-1}), \quad (3)$$

$$g_t = \tanh(U_g \cdot x_t + W_g \cdot h_{t-1}), \quad (4)$$

где $\sigma(z) = \frac{1}{1+e^{-z}}$ – сигмоидальная функция; $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ – функция гиперболического тангенса; x – входная последовательность; h – вектор скрытого состояния ячейки сети; $U_i, U_f, U_o, U_g, W_i, W_f, W_o, W_g$ – матрицы весовых коэффициентов фильтров сети i, f, o, g; индекс t – индекс элемента обучающей последовательности.

На основании значений фильтров сети определяются ее вектора внутреннего состояния (внутренней памяти) и скрытого состояния:

$$c_t = tah(i_t \circ g_t + f_t \circ c_{t-1}), \quad (5)$$

$$h_t = o_t \circ c_t, \quad (6)$$

где c_t - вектор внутреннего состояния ячейки сети; h_t – вектор скрытого состояния ячейки сети; $\circ$ - операция поэлементного произведения.

Уравнения (1) – (6) соответствуют этапу прямого распространения сигнала по сети. Структура ячейки LSTM сети представлена на рис. 2.

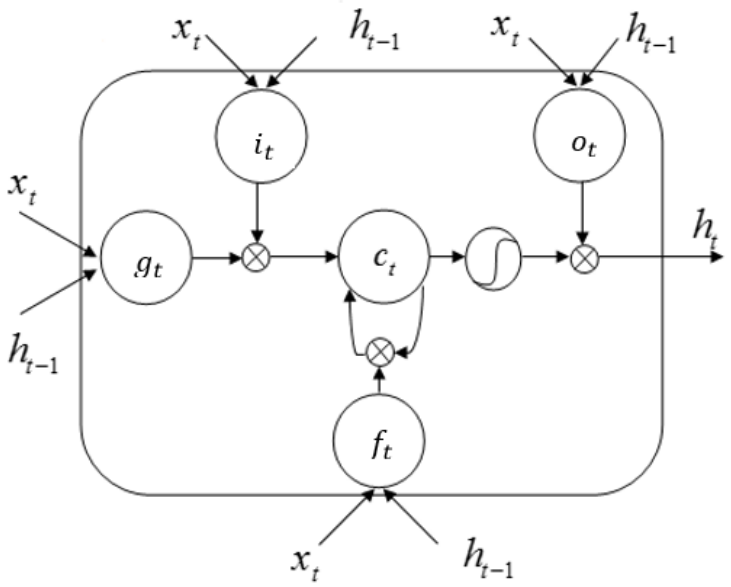

Рис. 2. Структура LSTM ячейки

Данный механизм позволяет LSTM сети бороться с проблемой исчезающего градиента при работе с длинными последовательностями. Посредством обучения параметров фильтров ($U_i, U_f, U_o, U_g, W_i, W_f, W_o, W_g$) сеть «настраивает» свою «память».

Несмотря на то, что LSTM сети спроектированы для борьбы с проблемой исчезающего градиента, значение вектора скрытого состояния ячейки LSTM может оказаться большим и, как следствие, привести к «насыщению» функции активации. Насыщением называется явление, при котором большие или малые значения, передаваемые на вход функции, приводят к ее выводу, близкому к пределу данной функции. Следствием этого являются малые значения производной функции и явление исчезающего градиента, которое значительно замедляет процесс обучения сети. И хотя LSTM сеть разработана для уменьшения влияния данного явления, она не может предотвратить его возникновение вследствие использования функции гиперболического тангенса в роли функции активации.

В данной работе для решения проблемы насыщаемости функции активации предложено использование функции активации, которая не насыщается. Данный подход позволит сделать обучение сети более быстрым и точным.

Функция активации, основанная на логарифме, изначально предложена в [5], позволяет избегать насыщения при обработке больших значений и определяется формулой:

$$f(x) = \begin{cases} \ln(x+1), x \geq 0, \\ -\ln(-x+1), x < 0 \end{cases}.$$

Производная данной функции определяется следующим образом:

$$\frac{\partial f}{\partial x} = \begin{cases} \frac{1}{1+x}, x \geq 0, \\ -\frac{1}{1-x}, x < 0. \end{cases}$$

Графики функции гиперболического тангенса, функции, основанной на логарифмах, и ее производной изображены на рис. 3.

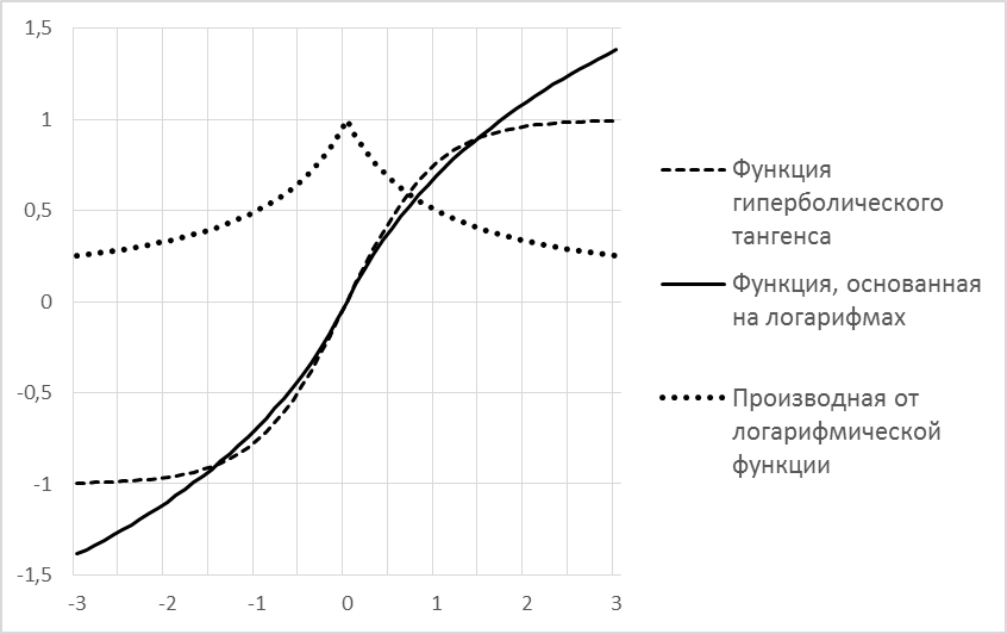

Рис. 3. График функции гиперболического тангенса и функции, основанной на логарифмах

Преимуществом предложенной функции активации по сравнению с функцией гиперболического тангенса является ее «ненасыщаемость» и потому ее применение улучшит эффективность обучения LSTM сети.

Сигмоидальная функция активации также подвержена проблеме насыщения, однако заменить ее предложенной не представляется возможным, поскольку она не может служить в качестве шлюза в силу того, что не масштабирует вводимые значения в интервале от 0 до 1.

В работе проверена возможность использования предложенной модели в задачах автоматического реферирования.

Методика

Для решения задачи моделирования текстов использована модель Encoder-Decoder. Изначально она была разработана в рамках создания системы машинного перевода. В ее основе заложена идея о преобразовании последовательности слов на одном языке в последовательность слов на другом, а задача заключается в определении наиболее вероятного перевода для входной последовательности слов. Применение данной модели к задаче автоматического реферирования оправданно, поскольку задачей реферирования также является преобразование исходного текста (последовательности слов и предложений) в реферат.

Модель Encoder-Decoder, описанная в [6], состоит из двух компонентов:

Энкодер считывает входную последовательность $x \in R^n$ и вычисляет ее векторное представление h_x. Декодер использует h_x для генерирования целевой последовательности $y' \in R^m$.

Схематичное изображение модели энкодер-декодер показано на рис. 4.

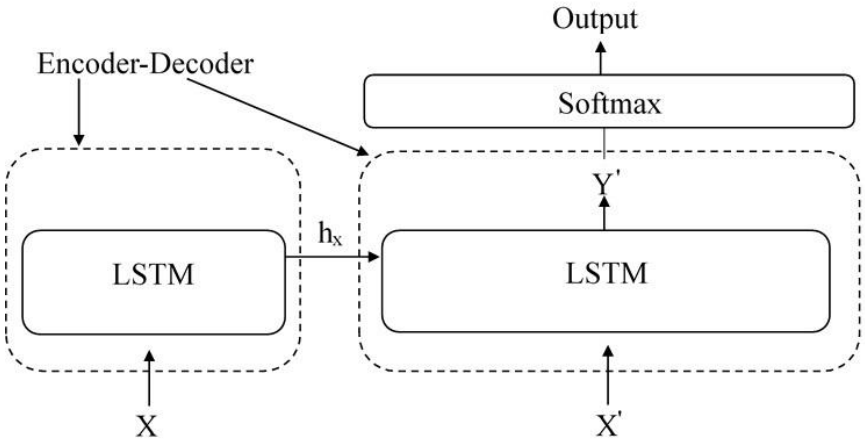

Рис. 4. Модель Encoder – Decoder

1 этап. Обучение

Шаг 1. Энкодер считывает исходное предложение $(x_1, x_2, ..., x_T)$.

Шаг 2. Вычисление векторного представления введенного предложения (вывод энкодера) и его передача на вход декодера.

Шаг 3. Передача вывода энкодера и целевого предложения $(x'_1, x'_2, ..., x'_{T-1})$ без последнего элемента на вход декодера.

Шаг 4. Определение вывода декодера. Для каждого слова целевого предложения декодером выводится вектор, элементы которого соответствуют вероятности появления каждого слова в словаре.

Шаг 5. Вычисление ошибки модели следующим образом: целевая последовательность, сдвинутая на один элемент вправо (предложение без маркера начала, но с маркером конца), поэлементно сравнивается с выводом декодера. Ошибка определяется по формуле перекрестной энтропии: $E(\hat{y}, y) = -\frac{1}{N} \cdot \sum_{n \in N} \hat{y}_n \cdot \log(y_n)$, где $\hat{y}$ – целевая последовательность, верное предложение; y - вывод декодера, предложенная моделью последовательность $(y'_1, y'_2, ..., y'_T)$.

2 этап. Тестирование обученной модели. На данном этапе шаги 1, 2 и 5 совпадают с шагами первого этапа, на шагах 3 и 4 выполняются действия:

Шаг 3. Декодер на вход принимает только векторное представление исходного предложения

Шаг 4. Генерация вывода декодера до тех пор, пока не будет сгенерирован маркер конца предложения <EOS>.

Важным моментом является различие длин исходной и целевой последовательностей.

Таким образом, модель принимает на вход исходное предложение, вычисляет его векторное представление и, на основании этого представления, генерирует предложение.

В качестве рабочих компонентов как энкодера, так и декодера используются LSTM сети. LSTM-энкодер на выходе имеет вектор фиксированной длины – векторное представление h_x входной последовательности $(x_1, x_2, ..., x_T)$. Этим вектором инициализируется скрытое состояние LSTM-декодера, который выполняет роль языковой модели [7] и, в свою очередь, выводит последовательность $(y'_1, y'_2, ..., y'_{T'})$. Данная последовательность поступает в слой нормализованной экспоненциальной функции (softmax), который позволяет получить вероятность итоговой последовательности.

Экспериментальная часть

Для подтверждения эффективности предложенной модификации проведен ряд экспериментов по сравнению базовых характеристик модели с использованием стандартной модели Encoder-Decoder и ее модификации [8]: зависимость времени обучения от объема словаря, от размерности скрытого слоя и от объема обучающего корпуса текстов.

1. Зависимость времени обучения от объема словаря

В данном эксперименте изменяется размерность словаря при постоянных значениях размерности скрытого слоя и объема обучающего корпуса текстов (100 и 10 соответственно). Результаты приведены в табл. 1.

Таблица 1

Зависимость времени обучения модели от количества слов в словаре

Количество слов в словаре	Время обучения, миллисекунды	
	стандартная модель Encoder-Decoder	модификация модели Encoder-Decoder
100	89,440	85,310
500	96,270	93,286
1000	105,748	100,885
3000	168,747	154,441
6000	226,566	203,110

2. Зависимость времени обучения от размерности скрытого слоя LSTM сети

В данном эксперименте изменяется размерность скрытого слоя, объем словаря и обучающего корпуса являются константными величинами: 100 и 100 слов соответственно. Результаты эксперимента приведены в табл. 2.

Таблица 2

Зависимость времени обучения от размерности скрытого слоя

Размерность скрытого слоя	Время выполнения для стандартного Encoder-Decoder, миллисекунды	Время выполнения для модифицированного Encoder-Decoder, миллисекунды
10	90,207	82,507
20	96,316	85,975
50	109,046	103,079
100	187,282	180,688
150	265,472	236,858

3. Зависимость времени обучения от объема обучающего корпуса текстов

В данном эксперименте переменной величиной является объем обучающего корпуса текстов, константными величинами являются объем словаря и размерность скрытого слоя: 100 и 10 соответственно. Результаты экспериментов приведены в табл. 3.

Таблица 3

Зависимость времени обучения от размера обучающего корпуса текстов

Количество слов в обучающем корпусе текстов	Время выполнения для стандартного Encoder-Decoder, миллисекунды	Время выполнения для модифицированного Encoder-Decoder, миллисекунды
100	1,063	0,9689
500	5,202	4,867
1000	10,464	9,833
3000	30,808	29,728
6000	61,624	58,103
8000	82,205	78,621

Исходя из табл. 1–3, можно сделать вывод о линейном характере полученных зависимостей.

Также графики отражают уменьшение времени выполнения алгоритма при использовании модифицированной модели. Это объясняется тем, что на вычисление функции гиперболического тангенса затрачивается больше операций. Полученные графики отражают тенденцию увеличения времени обучения модели в зависимости от изменяемых параметров.

Для сравнения скорости уменьшения ошибки в процессе обучения был проведен эксперимент по обучению модели с использованием модифицированной LSTM-сети и без модифицированной сети. Обучение выполнялось на одном и том же корпусе текстов, за одинаковое число эпох в обоих случаях. Процесс уменьшения ошибки в ходе обучения показан на рис. 5.

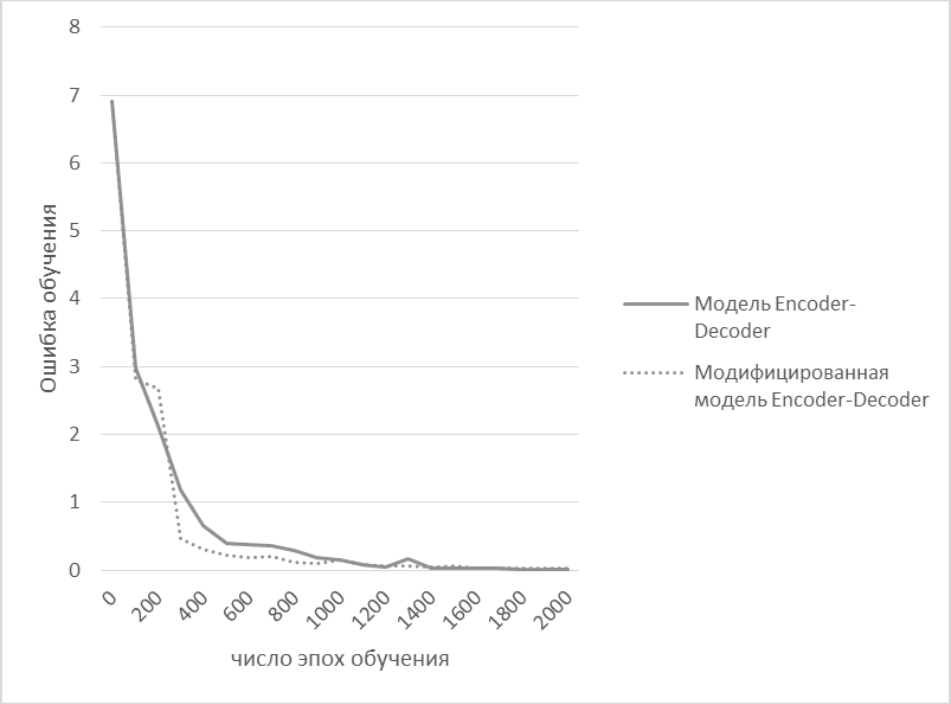

Рис. 5. График уменьшения ошибки модели в процессе обучения

На рисунке видно, что в процессе обучения ошибка уменьшается быстрее при использовании модифицированной модели Encoder-Decoder. Это объясняется тем, что функция, основанная на логарифмах активации, имеет большие значения производной при больших значениях вектора внутреннего состояния ячейки сети и вектора-кандидата нового внутреннего состояния ячейки сети.

Для экспериментальной проверки эффективности предложенной модели из материалов электронного новостного издания Science Daily [9] составлен корпус текстов объемом 30 статей. Тематика статей – песчаные смерчи на марсе (Mars Dust Devils). У каждой статьи использовался существующий реферат из нескольких предложений (1-6), рассматриваемый как эталонный для обучения модели.

С помощью разработанной программы на основе составленного корпуса обучена модель и построены рефераты для каждой статьи корпуса. Работа модели encoder-decoder оценивалась с помощью перечисленных метрик в сравнении с классическим алгоритмом на базе метода Луна. Для анализа результатов эффективности работы модели использовались численные метрики: точность π, полнота ρ и F-мера.

Также проведен эксперимент по оценке качества реферирования текста с использованием двух разных обученных моделей. Модели обучались на корпусах текстов разной тематики: Big Data и Mars Dust Devils. Объем словаря для корпусов составил 1053 и 995 слов со-

ответственно. Каждый корпус состоит из 15 текстов. Тематика текстов определялась на основе частотных характеристик важных слов. Для проверки качества реферата каждой модели на вход подан текст, наиболее близкий по своей теме к Mars Dust Devils.

Результаты

Усредненные результаты тестирования эффективности алгоритмов представлены в табл. 4.

Таблица 4

Усредненные оценки качества алгоритмов реферирования

Алгоритм реферирования	Численные метрики		
	Полнота	Точность	F-мера
Модифицированная модель Encoder-Decoder	0,862	0,735	0,791
Модель Encoder-Decoder	0,780	0,526	0,643
Метод Луна	0,670	0,353	0,523

Алгоритм с использованием предложенной модифиции модели Encoder-Decoder получил наивысшее среднее значение полноты (0,862), точности (0,730) и F-меры (0,768) в сравнении с другими рассмотренными алгоритмами. Самые низкие результаты показали рефераты, построенные с помощью классического алгоритма на основе метода Луна.

Выводы

В ходе данной работы исследована нейросетевая модель Encoder-Decoder в задаче автоматического реферирования текстов и предложена ее модификация с использованием ненасыщаемой функции активации, в качестве такой функции предложено использовать функцию активации, основанную на логарифмах. Результаты экспериментов показали эффективность выбранной модели и алгоритма классификации. Система показала лучший результат при совпадении тематики исходного текста и корпуса текстов, на котором обучена модель.

Разработана модификация модели Encoder-Decoder и выполнен сравнительный анализ полученной модели со стандартной моделью Encoder-Decoder и методом реферирования Луна. Модифицированная модель Encoder-Decoder может применяться в задачах автоматического реферирования, генерации текстов и машинного перевода.

Использование разработанной модели в совокупности с алгоритмами обработки текстовых данных является перспективным, так как это позволит автоматизировать процесс составления обучающих корпусов текстов, используемых для обучения системы реферирования.

В рамках дальнейшего развития описанного подхода предлагается обучение модели на больших объемах текстовых корпусов, создание нескольких моделей, обученных на корпусах определенной тематики.

Библиографический список

1. **Ломакина, Л.С.** Теоретические аспекты концептуального анализа и моделирования текстовых структур / Л.С. Ломакина, А.С. Суркова // Фундаментальные исследования. – 2015. – № 2 (часть 17). – С . 3713–3717.
2. **Bengio, Y.** Neural net language models. // Scholarpedia. 2008, 3(1) URL: http://www.scholarpedia.org/article/Neural_net_language_models.
3. **Хливненко, Л.В.** Практика нейросетевого моделирования / Л.В. Хливненко. – Воронеж: Воронежский государственный технический университет, 2015. – 214 с.
4. **Hochreiter, S.** Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural computation. 1997. 9(8):1735-1780.
5. **Bilski, J.** The backpropagation learning with logarithmic transfer function // Proceedings of the 5th Conference on Neural Networks and Soft Computing. 2000. – P. 71–76.

6. **Cho, K.** Learning phrase representations using RNN encoder-decoder for statistical machine translation / K. Cho [et al.] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing 2014. – P. 1724-1734.
7. **Sundermeyer, M.** LSTM Neural Networks for Language Modeling. / M. Sundermeyer, R. Schlüter, H. Ney / M. Sundermeyer, R. Schlüter, H. Ney // InInterspeech 2012. – P. 194–197.
8. **Суркова, А.С.** Использование модели Encoder-Decoder для реферирования текстов / А.С. Суркова, И.Д. Чернобаев // Системы управления и информационные технологии. – 2017. – №4(70). – С. 72–76.
9. Science Daily // Электронное новостное издание: сайт. – URL: https://www.sciencedaily.com (дата обращения 25.08.2017).

Дата поступления
в редакцию 15.01.2018

I. D. Chernobaev, A. S. Surkova, A.Z. Pankratova

TEXT MODELLING USING RECURRENT NEURAL NETWORKS

Nizhny Novgorod state technical university n.a. R.E. Alekseev

Purpose: Creation of model and algorithm to model natural language text with recurrent neural networks.
Design/methodology/approach: Application of recurrent neural networks, encoder-decoder architecture, non-saturating activation function.
Findings: Not-saturating activation function application allows achieving better results in model learning and exploitation.

*Key words***:** neural network, recurrent neural network, Long-Short-Term-Memory, activation function, natural language processing, Encoder-Decoder.