УДК 004.032.26

СОЗДАНИЕ ГЕНЕРАТОРА ПСЕВДОСЛОВ И КЛАССИФИКАЦИЯ ИХ СХОЖЕСТИ СО СЛОВАМИ СЛОВАРЯ РУССКОГО ЯЗЫКА МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

К. А. Ромаданский[1] [0009-0001-2912-3683], **А. Е. Ахаев**[2] [0009-0007-7154-2404], **Т. Р. Гилязов**[3] [0009-0009-9643-0200]

[1–3]*Казанский (Приволжский) федеральный университет, Институт информационных технологий и интеллектуальных систем*

[1]kirillaromad@mail.ru, [2]aeahaev@gmail.com, [3]t1g2r3mr@gmail.com

Аннотация

Под псевдословом понимается единица речи или текста, которая выглядит как реальное слово на русском языке, но на самом деле не имеет значения, а под настоящим или естественным словом – единица речи или текста, которая имеет толкование и представлена в словаре. Представлены две модели для работы с русским языком: генератор псевдослов и классификатор, оценивающий степень схожести введенной последовательности символов с настоящими словами. Классификатор использован для оценки результатов генератора. Обе модели основаны на рекуррентной нейронной сети с долгой краткосрочной памятью и обучены на датасете существительных русского языка. В результате создан файл, содержащий список сгенерированных псевдослов, оцененных классификатором. Псевдослова могут найти применение в задачах нейминга, брендирования и макетирования, в искусстве, для создания креативных произведений, и в языковых исследованиях, для изучения структуры языка и слов.

Ключевые слова*: генерация слов, псевдослово, нейронная сеть, рекуррентная нейронная сеть, долгая краткосрочная память*

ВВЕДЕНИЕ

Современные методы и алгоритмы машинного обучения позволяют строить модели на основе больших объемов данных [1, 2] и выявлять скрытые закономерности в них [3]. Это открывает возможности для создания интеллектуальных систем, способных генерировать новые данные [4]. Идеей настоящего исследования является создание модели для генерации псевдослов, которые будут схожи с естественными словами и соответствовать языковым правилам [5, 6].

Цель исследования – реализация генератора псевдослов. В качестве инструмента анализа и обработки реальных слов и выделения словообразовательных правил языка [7] применены нейронные сети. Пример такого правила – порядок букв в слове, допустимый с точки зрения морфологии языка.

Такой генератор позволяет создавать интересные и уникальные псевдослова, которые могут найти свое применение в различных областях, например:

- нейминг – генератор может применяться для генерации псевдонимов, названий компаний [8], проектов и т. д.;
- искусство – сгенерированные слова могут быть использованы в литературе, музыке и других формах искусства для создания уникальных и креативных произведений [9];
- макетирование – генератор может использоваться для заполнения текстом различных шаблонов и макетов;
- языковые исследования – для лингвистов и исследователей языка генератор может стать интересным инструментом для изучения структуры языка и слов [10].

Целевой язык исследования – русский, часть речи – существительные.

ИНСТРУМЕНТЫ РЕАЛИЗАЦИИ

Отличие текста от других типов данных (например, изображений) состоит в наличии временной компоненты. Таким же свойством обладают звук и временные ряды. Текст читается не моментально, а символ за символом, в строго определенном порядке. Поэтому использовалась нейросеть, которая учитывает эту особенность текста [11].

Наиболее подходящим инструментом реализации является рекуррентная нейронная сеть (англ. recurrent neural network, RNN) [12]. Этот вид нейронных сетей обладает внутренней памятью, что позволяет учитывать предыдущие состояния при обработке текущего входа. При реализации использовалась разновидность рекуррентных нейронных сетей с долгой краткосрочной памятью (англ. long short-term memory, LSTM). Для построения модели нейронной сети применена библиотека Keras для языка программирования Python [13].

РЕАЛИЗАЦИЯ ГЕНЕРАТОРА НОВЫХ СЛОВ

Датасет

В качестве исходного набора данных был выбран датасет “Russian Dictionary Data” из существительных русского языка объемом около 27000 слов [14]. Из них для обучения модели генератора псевдослов были выбраны слова длиной от 5 до 10 символов. Также особенностью отобранных существительных является отсутствие дефисов и повторяющихся букв, идущих друг за другом. Отфильтрованный датасет для генератора содержал 18283 слова.

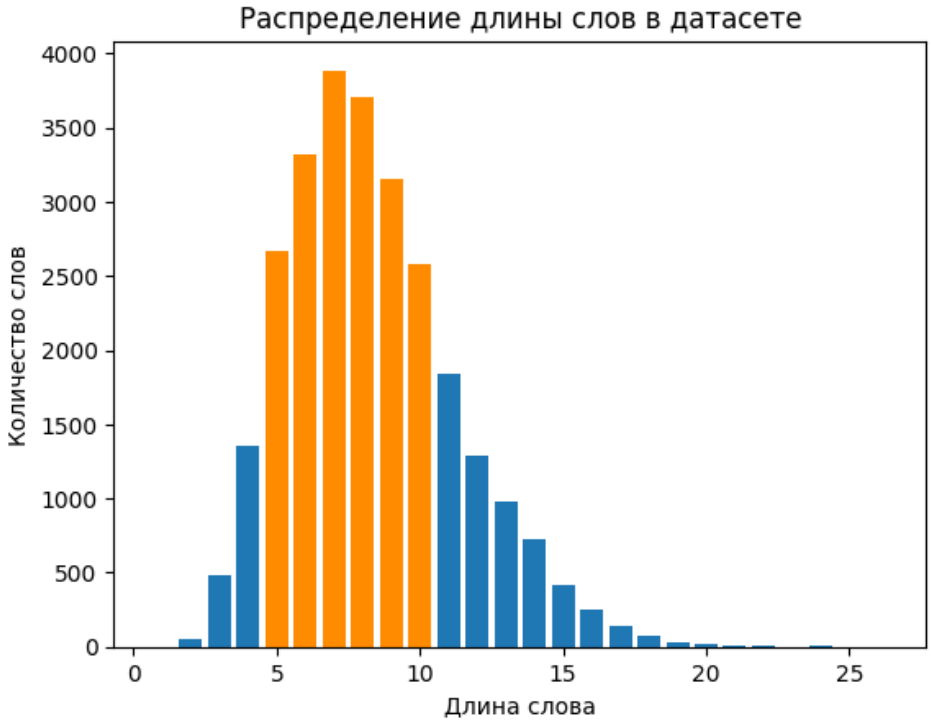

Рис. 1. Распределение длин слов в исходном датасете и слов, отобранных для обучения генератора псевдослов

На Рис. 1 проиллюстрировано распределение длин слов исходного датасета, где оранжевым выделены отобранные по длине слова для обучения генератора.

Подготовка данных

Для каждого отобранного слова из датасета был сформирован набор всевозможных пар начала и конца слова. Например, пары для слова “кот”:

- “к” и “от”;
- “ко” и “т”.

Сформированная коллекция начал слов использовалась в качестве входного набора данных для обучения модели генератора псевдослов, а коллекция концов – выходного набора данных.

Для передачи входных данных в модель генератора использовалась векторизация строк, т. е. их математическое представление в виде матриц. Для этого применен подход one-hot кодирования [15].

Так как набор возможных символов строк содержал 34 элемента (33 буквы алфавита и 1 дополнительный пустой символ), а максимальная длина строки равна 9 (максимальная длина слова в датасете минус 1), то каждая входная строка кодировалась в виде матрицы размерностью 9 на 34. В случае, если длина строки меньше максимальной длины, строка дополнялась пустыми символами. Пример векторизации строки «чело», являющейся началом слова «человек», изображен на Рис. 2.

```
    а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
(9, 34)
```

Рис. 2. Векторное представление входных данных модели генератора псевдослов на примере строки «чело», являющейся началом слова «человек»

Архитектура модели

Модель генератора псевдослов построена на основе пятислойной нейронной сети с использование LSTM слоев, где в качестве функции потерь использована категориальная кросс-энтропия [16]. Схематично архитектура модели представлена на Рис. 3.

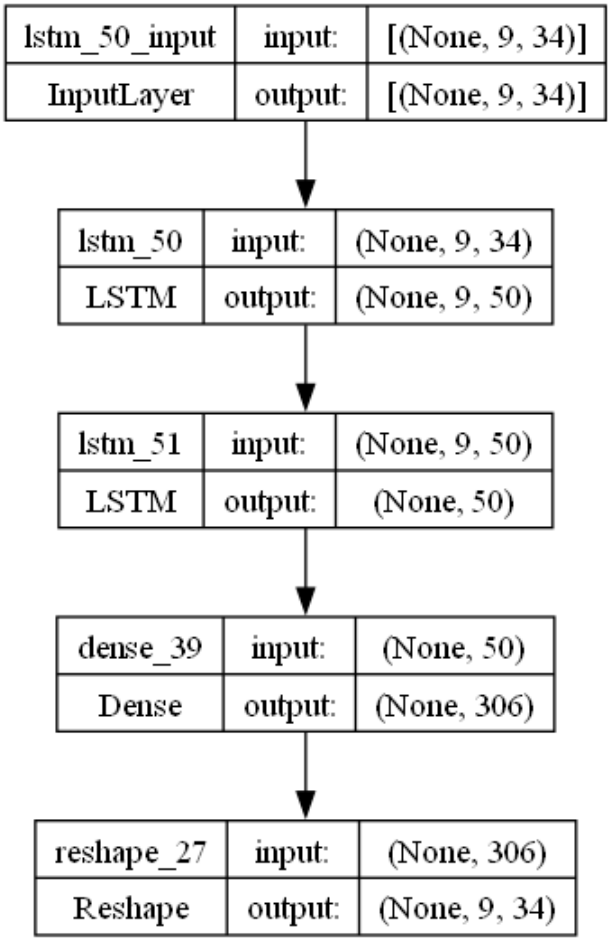

Рис. 3. Архитектура нейронной сети генератора псевдослов

Модель генератора состоит из следующих слоев:

1. входной слой;
2. первый LSTM-слой;
3. второй LSTM-слой;
4. полносвязный слой;
5. выходной слой.

Выходом модели является матрица той же размерности, что и входная матрица. Это и есть сгенерированное продолжение входной последовательности

символов. Конкатенация входной и выходной строк модели является сгенерированным псевдословом.

Обучение модели

Результатом работы модели должно быть сгенерированное псевдослово, соответствующее грамматическим правилам языка и похожее на настоящее, но не являющееся им. Найденное решение – построение не слишком “точной” модели, т. е. допускаются незначительное переобучение и не самая лучшая точность (accuracy) модели.

Для обучения сгенерированный набор входных и выходных данных был разделен на обучающую и тестовую выборки в соотношении 9 к 1.

Обучение модели продолжалось до тех пор, пока на протяжении пяти эпох подряд не увеличивалось значение метрики accuracy. Процесс обучения завершился на 165-й эпохе. При дальнейшем обучении наблюдались сильное переобучение и падение точности, что представлено на Рисунках 4 и 5.

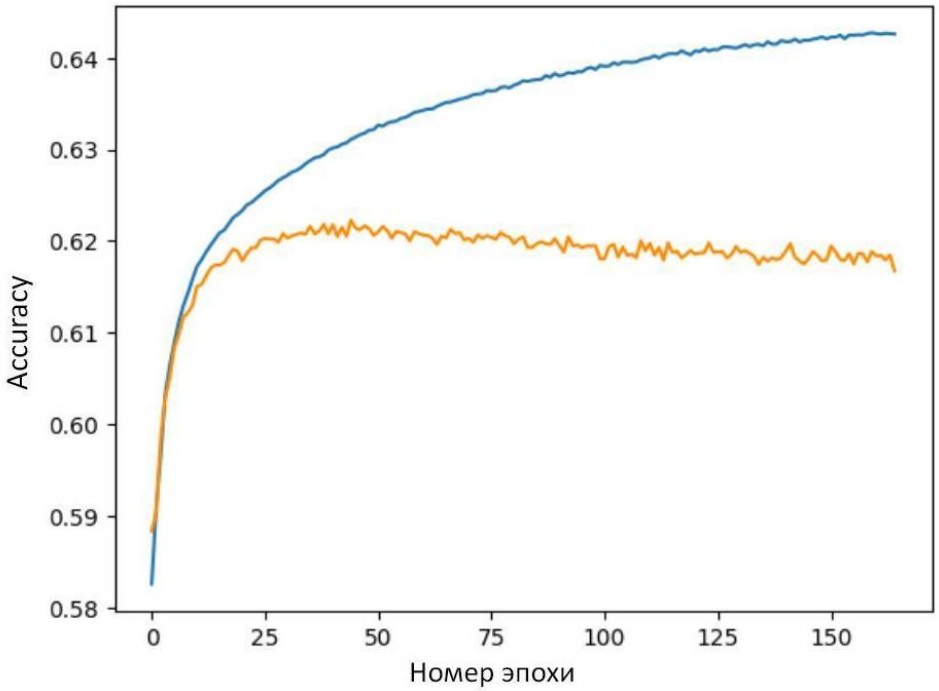

Рис. 4. Графики точности (accuracy) обученной модели генератора на обучающей (синий) и тестовой (оранжевый) выборках

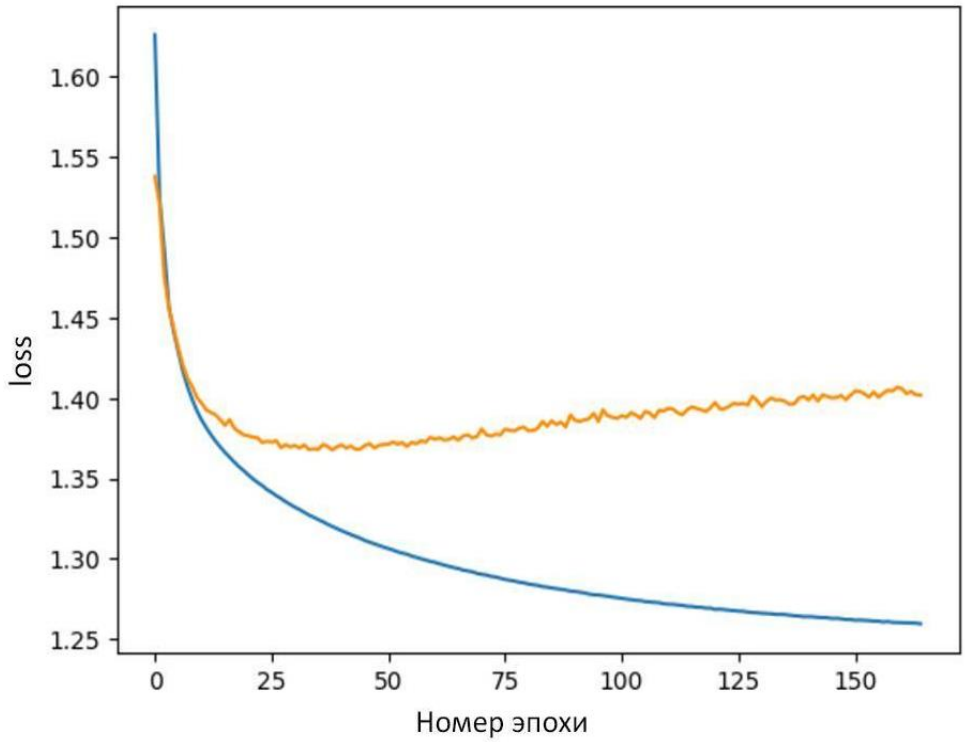

Рис. 5. Графики функции потерь (loss-функции) обученной модели генератора на обучающей (синий) и тестовой (оранжевый) выборках

РЕАЛИЗАЦИЯ КЛАССИФИКАТОРА РЕАЛЬНОСТИ СЛОВ

Датасет

В качестве исходного набора данных использовался тот же датасет “Russian Dictionary Data”, что и для генератора псевдослов. Однако для обучения модели классификатора отобраны слова без дефисов длиной от 3 до 15 символов. Количество таких слов составило 26084 (Рис. 6).

Также для обучения классификатора были созданы случайно сгенерированные последовательности кириллических символов в том же количестве. Так, итоговый размер датасета для классификатора составил 52168 строк.

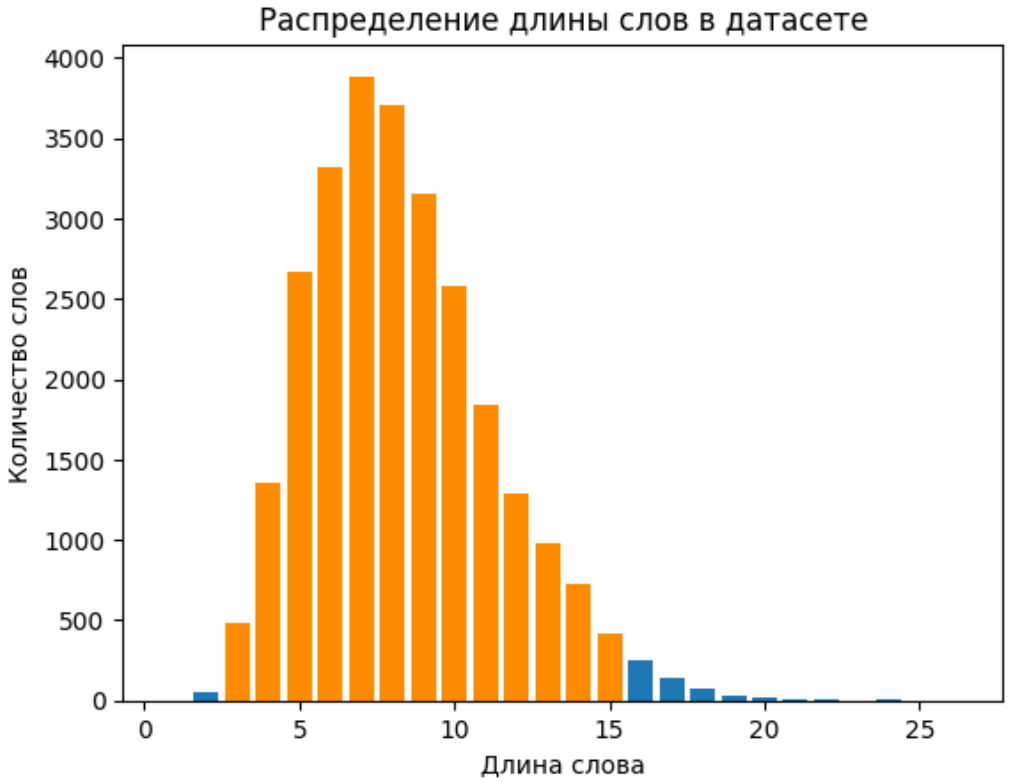

Рис. 6. Распределение длин слов в исходном датасете и слов, отобранных для обучения классификатора слов

Подготовка данных

Для обучения классификатора каждое настоящее слово было помечено значением 1, а сгенерированная последовательность символов – 0.

Над входными строками была проведена векторизация для передачи их в модель классификатора по аналогии с моделью генератора. Математическим представлением строк является матрица размерностью 15 на 33, где 15 – это максимальная длина слова в датасете, а 33 – размер русского алфавита. Слова длиной меньшей 15 букв дополнялись до 15 пустыми символами.

Архитектура модели

Модель классификатора построена на основе трехслойной нейронной сети (входной, LSTM и выходной слои). В качестве функции потерь использована бинарная кросс-энтропия, а в качестве основной метрики – accuracy [17]. Архитектура полученной модели классификатора представлена на Рис. 7.

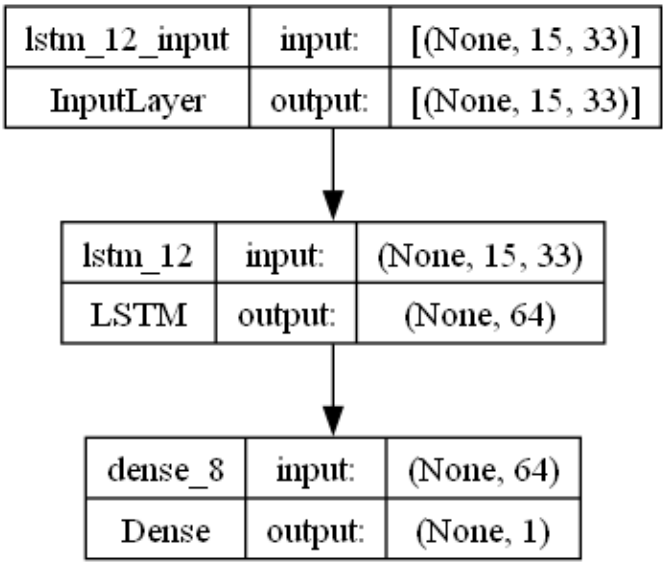

Рис. 7. Архитектура нейронной сети классификатора слов

Выход модели – число от 0 до 1; чем ближе оно к 1, тем больше входная последовательность символов похожа на настоящее слово.

Обучение модели

Для обучения сгенерированный набор входных и выходных данных разделен на обучающую и тестовую выборки в соотношении 8 к 2.

Эмпирически выявлено, что для обучения классификатора достаточно 15 эпох. Итоговое значение accuracy составило 0.9779, функции потерь – 0.0604. На рисунках 8 и 9 представлены соответствующие графики accuracy и функции потерь модели классификатора во время обучения.

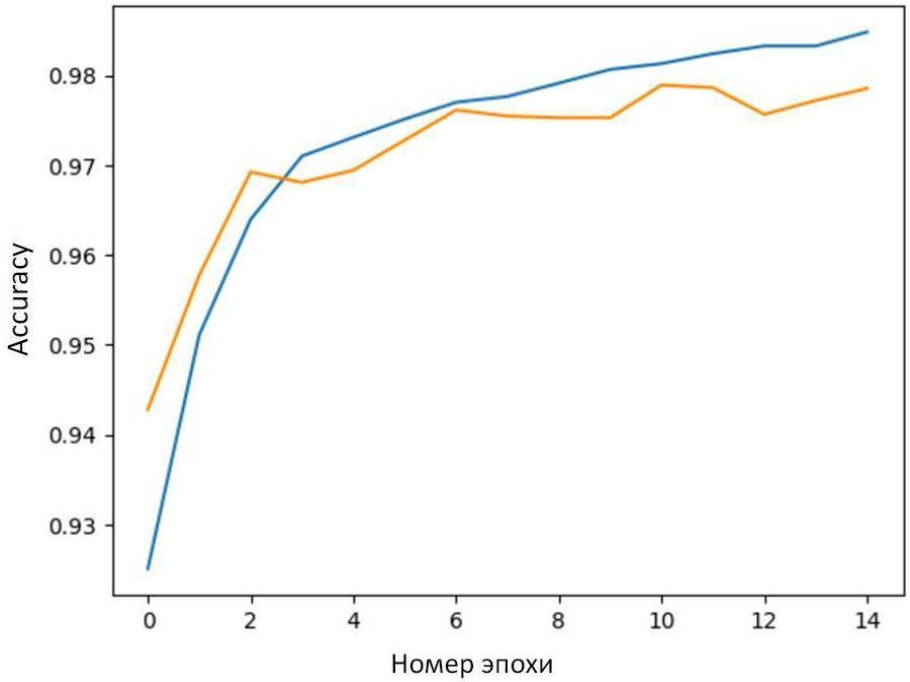

Рис. 8. Графики accuracy обученной модели классификатора слов на обучающей (синий) и тестовой (оранжевый) выборках

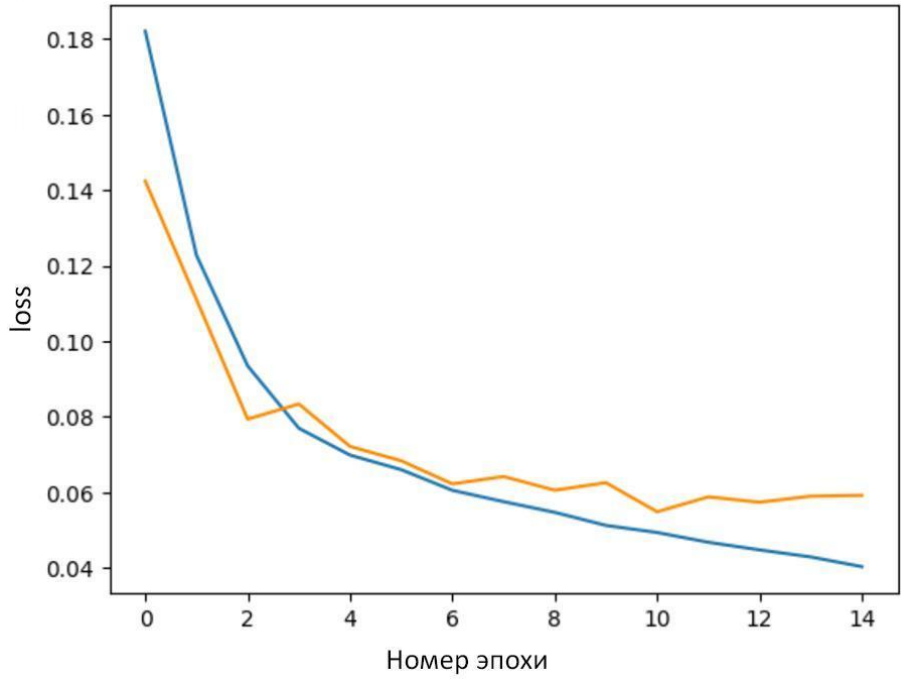

Рис. 9. Графики loss-функции обученной модели классификатора слов на обучающей (синий) и тестовой (оранжевый) выборках

ГЕНЕРАЦИЯ ПСЕВДОСЛОВ

Для генерации псевдослов сформирован массив входных данных для модели генератора. В этот массив отобраны уникальные начала слов из всего исходного датасета длиной от 1 до 3 символов. В результате сгенерировано 3271 слово. Для сохранения результатов сформирован список объектов, содержащих следующие свойства:

1. Input – входная последовательность символов для генератора.
2. Output – выходная последовательность символов генератора.
3. Result – конкатенация значений Input и Output. Полученная последовательность является новым псевдословом.

Полученный массив объектов был сохранен в файл формата JSON. Пример JSON-объектов, записанных в файл, представлен на Рис. 10.

Далее, сгенерированные псевдослова были оценены классификатором. Полученные оценки также были сохранены в JSON-файл. Для этого к свойствам Input, Output и Result было добавлено свойство Value, содержащее оценку строки Result классификатором (Рис. 11).


```
{
  "Input": "пус",
  "Output": "ток",
  "Result": "пусток"
},
{
  "Input": "пут",
  "Output": "ер",
  "Result": "путер"
},
{
  "Input": "пух",
  "Output": "овка",
  "Result": "пуховка"
},
{
  "Input": "зип",
  "Output": "ун",
  "Result": "зипун"
},
{
  "Input": "зия",
  "Output": "ние",
  "Result": "зияние"
},
{
  "Input": "зл",
  "Output": "ова",
  "Result": "злова"
},
{
  "Input": "омш",
  "Output": "етка",
  "Result": "омшетка"
},
{
  "Input": "омы",
  "Output": "зение",
  "Result": "омызение"
},
{
  "Input": "омё",
  "Output": "т",
  "Result": "омёт"
},
```


Рис. 10. Фрагмент результатов работы генератора, сохраненных в JSON-файле, где Input – входная последовательность символов, Output – выходная последовательность и Result – получившееся псевдослово

```
{
  "Input": "мещ",
  "Output": "анк",
  "Result": "мещанк",
  "Value": 0.984096109867096
},
{
  "Input": "мз",
  "Output": "йъои",
  "Result": "мзйъои",
  "Value": 2.96324178634677708e-05
},
{
  "Input": "мзд",
  "Output": "оеа",
  "Result": "мздоеа",
  "Value": 0.16401644051074982
},
{
  "Input": "пюр",
  "Output": "ати",
  "Result": "пюрати",
  "Value": 0.9994021654129028
},
{
  "Input": "пя",
  "Output": "тоаь",
  "Result": "пятоаь",
  "Value": 0.10005672812461853
},
{
  "Input": "пяд",
  "Output": "ани",
  "Result": "пядани",
  "Value": 0.999237596988678
},
{
  "Input": "рь",
  "Output": "ано",
  "Result": "рьано",
  "Value": 0.8694648742675781
},
{
  "Input": "рья",
  "Output": "ниа",
  "Result": "рьяниа",
  "Value": 0.9554255604743958
},
{
  "Input": "рэ",
  "Output": "каа",
  "Result": "рэкаа",
  "Value": 0.4970543384552002
},
```


Рис. 11. Фрагмент результатов работы генератора и классификатора, сохраненных в JSON-файле, где Input – входная последовательность символов, Output – выходная последовательность, Result – получившееся псевдослово и Value – оценка реальности псевдослова классификатором

Оценка генератора была использована для выявления и фильтрации сгенерированных результатов, не похожих на настоящие слова. В качестве нижнего приемлемого значения оценки классификатора выбрано число 0.9. Пример слов, не подходящих под минимальную оценку, выделен желтым цветом на Рис. 11. В результате были выделены сгенерированные псевдослова, оценка классификатором которых не ниже 0.9. Таких слов оказалось 2778, что составляет примерно 85% от всего количества сгенерированных псевдослов.

ЗАКЛЮЧЕНИЕ

Созданы две модели: генератор псевдослов, похожих на естественные слова русского языка, и классификатор, оценивающий, насколько последовательность символов похожа на слово русского языка. Получившиеся псевдослова могут быть полезны для разработки названий и брендов, создания макетов, креативных проектов, в сфере искусства, а также в языковых исследованиях, направленных на изучение структуры языка и слов.

Архитектура генератора представляет пятислойную нейронную сеть с двумя LSTM-слоями. На вход генератор принимает пользовательский ввод (начало

слова), а на выходе возвращает предлагаемое продолжение слова. Итоговое псевдослово получается путём конкатенации пользовательского ввода и сгенерированного продолжения. Для обучения модели генератора использован датасет существительных русского языка с ограничением по длине от 5 до 10 символов без использования слов с подряд идущими одинаковыми буквами. Обучение модели продолжалось до тех пор, пока в течение пяти последовательных эпох не наблюдалось увеличение значения метрики accuracy, и было остановлено на 165-й эпохе.

Классификатор представляет трехслойную нейронную сеть с одним LSTM-слоем. На вход классификатор принимает пользовательский ввод, то есть любую последовательность кириллических символов, а возвращает число от 0 до 1 – степень схожести введенной последовательности символов со словом русского языка. Для обучения модели классификатора также использовался датасет существительных русского языка длиной от 3 до 15 символов. Число эпох для обучения модели классификатора было выявлено эмпирически. Для достижения оптимальных результатов потребовалось 15 эпох.

Для тестирования функционала созданных моделей из датасета существительных были отобраны уникальные начала слов. Они были использованы как вход для модели генератора. Модель классификатора использовалась для оценки реальности каждого из сгенерированных слов. Итого получилось 3271 псевдослово, 85% из которых оценены классификатором на 0.9 и выше.

Результаты работы представлены в открытом доступе в репозитории GitHub: https://github.com/smtyper/word-craft-ai.

БЛАГОДАРНОСТИ

Выражаем благодарности Липачёву Евгению Константиновичу и Елизарову Александру Михайловичу за проявленный интерес к исследованию, значимые замечания и советы при оформлении статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Sagiroglu S., Sinanc D.* Big Data: A Review // 2013 International Conference on Collaboration Technologies and Systems (CTS). 2013. P. 42–47.

2. *Shim K.* MapReduce algorithms for Big Data Analysis // Proceedings of the VLDB Endowment. 2012. V. 5. No. 12. P. 2016–2017.

3. *Строев В.В., Тихонов А.И.* Применение технологий Data Mining для поиска соответствий закономерностей развития в больших массивах веб-данных на основе инструментов анализа Big Data // E-Management. 2022. Т. 5. N 4. С. 4–11.

4. *Kim J., Shin S., Bae K., Oh S.* Can AI be a content creator? Effects of content creators and information delivery methods on the psychology of content consumers // Telematics and Informatics. 2020. V. 55. P. 101452.

5. *Лалетина А.О.* Языковая норма в эпоху глобализации // Ученые записки Казанского университета. Серия Гуманитарные науки. 2011. Т. 153. № 6. С. 219–226.

6. *Москалёва М.В.* Неологизмы и проблема их изучения в современном русском языке // Известия РГПУ им. А. И. Герцена. 2008. № 80. С. 246–250.

7. *Дмитриева Д.Д.* Изучение словообразования на занятиях по русскому языку как иностранному // Балтийский гуманитарный журнал. 2020. Т. 9. № 1(30). С. 47–49.

8. *Shipley D., Hooky G.J., Wallace S.* The brand name Development Process // International Journal of Advertising. 1988. V. 7. No. 3. P. 253–266.

9. *Mazzola G., Carapezza M., Chella A., Mantoan D*. Artificial Intelligence in Art Generation: An Open Issue // Image Analysis and Processing – ICIAP 2023 Workshops. 2023. V. 14366. P. 258–269.

10. *Jarmulowicz L., Taran V.L.* Lexical morphology // Topics in Language Disorders. 2013. V. 33. No. 1. P. 57–72.

11. *Iqbal T., Qureshi S.* The survey: Text generation models in deep learning // Journal of King Saud University - Computer and Information Sciences. 2022. V. 34. No. 6. P. 2515–2528.

12. *Yu Y., Si X., Hu C., Zhang J.* A review of Recurrent Neural Networks: LSTM cells and network architectures // Neural Computation. 2019. Т. 31. No. 7. P. 1235–1270.

13. *Ketkar N.* Introduction to Keras // Deep Learning with Python. Berkeley, CA: Apress, 2017. P. 97–111.

14. *Helms M.* Badestrand/Russian-Dictionary: Dataset of nouns, verbs, adjectives and others from my Russian dictionary website OpenRussian.org. [Электронный

ресурс]. URL: https://github.com/Badestrand/russian-dictionary (дата обращения: 17.10.2023).

15. *Rodríguez P., Bautista M.A., Gonzàlez J., Escalera S.* Beyond one-hot encoding: Lower dimensional target embedding // Image and Vision Computing. 2018. V. 75. P. 21–31.

16. *Mao A., Mohri M., Zhong Y*. Cross-entropy loss functions: Theoretical analysis and applications // Proceedings of the 40th International Conference on Machine Learning. 2023. V. 202. P. 23803–23828.

17. *Manaswi N.K.* Understanding and Working with Keras // Deep Learning with Applications Using Python. Berkeley, CA: Apress, 2018. P. 31–43.

CREATING PSEUDOWORDS GENERATOR AND CLASSIFIER OF THEIR SIMILARITY WITH WORDS FROM RUSSIAN DICTIONARY USING MACHINE LEARNING

K. A. Romadanskiy[1] [0009-0001-2912-3683], A. E. Akhaev[2] [0009-0007-7154-2404],

T. R. Gilyazov[3] [0009-0009-9643-0200]

[1–3]*Kazan (Volga region) Federal University, Institute of Information Technology and Intelligent Systems.*

[1]kirillaromad@mail.ru, [2]aeahaev@gmail.com, [3]t1g2r3mr@gmail.com

Abstract

In this article, a pseudoword is defined as a unit of speech or text that appears to be a real word in Russian but actually has no meaning. A real or natural word is a unit of speech or text that has an interpretation and is presented in a dictionary. The paper presents two models for working with the Russian language: a generator that creates pseudowords that resemble real words, and a classifier that evaluates the degree of similarity between the entered sequence of characters and real words. The classifier is used to evaluate the results of the generator. Both models are based on recurrent neural networks with long short-term memory layers and are trained on a dataset of Russian nouns. As a result of the research, a file was created containing a list of pseudowords generated by the generator model. These words were then evaluated by the classifier to filter out those that were not similar enough to real words. The

generated pseudowords have potential applications in tasks such as name and branding creation, layout design, art, crafting creative works, and linguistic studies for exploring language structure and words.

Keywords*: word generation, pseudoword, neural network, recurrent neural network, long short-term memory*

REFERENCES

1. *Sagiroglu S., Sinanc D.* Big Data: A Review // 2013 International Conference on Collaboration Technologies and Systems (CTS). 2013. P. 42–47.

2. *Shim K.* MapReduce algorithms for Big Data Analysis // Proceedings of the VLDB Endowment. 2012. T. 5. No. 12. P. 2016–2017.

3. *Stroev V.V., Tikhonov A.I.* Application of data mining technologies to find correspondences of development patterns in large arrays of web data based on Big Data Analysis Tools // E-Management. 2022. V. 5. No. 4. P. 4–11.

4. *Kim J., Shin S., Bae K., Oh S.* Can AI be a content creator? Effects of content creators and information delivery methods on the psychology of content consumers // Telematics and Informatics. 2020. V. 55. P. 101452.

5. *Laletina A.O*. Yazykovaya norma v epohu globalizacii // Uchenye zapiski Kazanskogo universiteta. Seriya Gumanitarnye nauki. 2011. V. 153. No. 6. P. 219–226

6. *Moskalyova M. V.* Neologizmy i problema ih izucheniya v sovremennom russkom yazyke // Izvestia: Herzen University Journal of Humanities & Sciences. 2008. No. 80. P. 246–250.

7. *Dmitrieva D.D.* Izuchenie slovoobrazovaniya na zanyatiyah po russkomu yazyku kak inostrannomu // Baltijskij gumanitarnyj zhurnal. 2020. V. 9. No. 1(30). P. 47–49.

8. *Shipley D., Hooky G.J., Wallace S.* The brand name Development Process // International Journal of Advertising. 1988. V. 7. No. 3. P. 253–266.

9. *Mazzola G., Carapezza M., Chella A., Mantoan D.* Artificial Intelligence in Art Generation: An Open Issue // Image Analysis and Processing – ICIAP 2023 Workshops. 2023. P. 14366. V. 258-269.

10. *Jarmulowicz L., Taran V.L.* Lexical morphology // Topics in Language Disorders. 2013. V. 33. No. 1. P. 57–72.

11. *Iqbal T., Qureshi S.* The survey: Text generation models in deep learning // Journal of King Saud University - Computer and Information Sciences. 2022. V. 34. No. 6. P. 2515–2528.

12. *Yu Y., Si X., Hu C., Zhang J.* A review of Recurrent Neural Networks: LSTM cells and network architectures // Neural Computation. 2019. T. 31. No. 7. P. 1235–1270.

13. *Ketkar N.* Introduction to Keras // Deep Learning with Python. Berkeley, CA: Apress, 2017. C. 97–111.

14. *Helms M.* Badestrand/Russian-Dictionary: Dataset of nouns, verbs, adjectives and others from my Russian dictonary website OpenRussian.org. URL: https://github.com/Badestrand/russian-dictionary (accessed: 17.10.2023).

15. *Rodríguez P., Bautista M.A., Gonzàlez J., Escalera S.* Beyond one-hot encoding: Lower dimensional target embedding // Image and Vision Computing. 2018. V. 75. P. 21–31.

16. *Mao A., Mohri M., Zhong Y*. Cross-entropy loss functions: Theoretical analysis and applications // Proceedings of the 40th International Conference on Machine Learning. 2023. V. 202. P. 23803–23828.

17. *Manaswi N.K.* Understanding and Working with Keras // Deep Learning with Applications Using Python. Berkeley, CA: Apress, 2018. P. 31–43.

СВЕДЕНИЯ ОБ АВТОРАХ

РОМАДАНСКИЙ Кирилл Алексеевич – магистрант Казанского Федерального Университета.

Kirill Alekseevich ROMADANSKIY – master student of Kazan Federal University.

email: kirillaromad@mail.ru

ORCID: 0009-0001-2912-3683

АХАЕВ Артемий Евгеньевич – магистрант Казанского Федерального Университета.

Artemii Evgenyevich AKHAEV – master student of Kazan Federal University.

email: aeahaev@gmail.com

ORCID: 0009-0007-7154-2404

ГИЛЯЗОВ Тагмир Радикович – магистрант Казанского Федерального Университета.

Tagmir Radikovich GILYAZOV – master student of Kazan Federal University.

email: t1g2r3mr@gmail.com

ORCID: 0009-0009-9643-0200

Материал поступил в редакцию 25 ноября 2024 года