ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2020 Управление, вычислительная техника и информатика № 50

ОБРАБОТКА ИНФОРМАЦИИ

УДК 004.855
DOI: 10.17223/19988605/50/2

И.А. Батраева, А.Д. Нарцев, А.С. Лезгян

ИСПОЛЬЗОВАНИЕ АНАЛИЗА СЕМАНТИЧЕСКОЙ БЛИЗОСТИ СЛОВ ПРИ РЕШЕНИИ ЗАДАЧИ ОПРЕДЕЛЕНИЯ ЖАНРОВОЙ ПРИНАДЛЕЖНОСТИ ТЕКСТОВ МЕТОДАМИ ГЛУБОКОГО ОБУЧЕНИЯ

Рассматриваются вопросы применения сверточных нейронных сетей для анализа текстов с точки зрения определения их жанровой принадлежности. Описана разработанная архитектура сверточной нейронной сети с использованием векторного представления слов на основе модели word2vec, приведены результаты экспериментов по обучению сети.
Ключевые слова: машинное обучение; сверточные нейронные сети; модель word2vec; интеллектуальный анализ текстов.

Автоматизация извлечения различной информации из текстов стала одной из основных проблем, связанных с информационным поиском. Так как тексты чаще всего либо являются слабо структурированными, либо вообще не обладают структурой с точки зрения решаемой задачи, то особо важным стало направление интеллектуального анализа текстов, включающее в себя методы классификации и анализа текстов на основе алгоритмов машинного обучения.

В частности, одной из задач анализа текстов является тематическая классификация, которая позволяет определить принадлежность текста к определенной группе тем. Особенно актуальна такая классификация для решения задач корпусной лингвистики, так как в большинстве существующих на сегодняшний день корпусов деление по темам и жанрам выполняется вручную или исходя из тематики источников текста [1]. Особенностью классификации текстов языковых корпусов является то, что для них важна, скорее, литературная классификация по темам (война, история, фантастика, сказки и т.д.) и жанрам (песни, стихи, повествование и т.п.). В данной работе рассматривается решение задачи классификации текстов для использования в языковых корпусах.

Более формально задача жанровой классификации может быть сформулирована так: даны текст на естественном языке и множество возможных жанров, к которым он может принадлежать. Требуется определить жанр текста. Если текст относится к нескольким жанрам одновременно, то определить основной жанр.

В последнее время наибольшую популярность для решения задач классификации приобрели глубокие нейронные сети, так как они позволяют достичь наивысшей точности среди всех известных моделей машинного обучения. В частности, сверточные нейронные сети совершили прорыв в классификации изображений. В настоящее время они успешно справляются и с некоторыми задачами автоматической обработки текстов. Более того, как утверждается в некоторых исследованиях [2–5], сверточные сети подходят для этого даже лучше рекуррентных нейронных сетей, которые чаще всего используются для анализа текстовых последовательностей [6]. С другой стороны, использование сверточных сетей для классификации текстов мало исследовано. Поэтому исследование применения сверточных нейронных сетей для задачи классификации текстов в качестве альтернативы рекуррентным нейронным сетям представляет практический интерес.

Для решения поставленной задачи требуется получить способ представления данных в виде, пригодном для обработки сверточной нейронной сетью. Например, в виде матрицы вещественных чисел. Наиболее распространенным является способ отображения каждого слова в многомерное векторное пространство. В рамках данной работы векторные представления слов строились на основе модели word2vec [7].

Таким образом, поставленная задача решалась сверточной нейронной сетью, на вход которой подавались векторные представления слов, полученные при обучении модели word2vec.

1. Предварительная обработка данных

На вход модели должен подаваться заранее обработанный корпус текстов. Предварительная обработка состоит из следующих этапов:

– Удаление всех знаков препинания, чисел и слов «нецелевых» языков (не предназначенных для обработки моделью).

– Разбиение текста на предложения. Для этого был выбран пакет библиотек Natural Language Toolkit (NLTK). Данная библиотека применяет регулярные выражения, а также некоторые алгоритмы машинного обучения для обработки естественного языка. Базовая версия NLTK не поддерживает разбиение русскоязычных текстов на предложения, поэтому использовалась модификация, расширяющая функционал библиотеки [8, 9].

– Удаление «стоп-слов» – слов, не несущих определенной смысловой нагрузки, но при этом затрудняющих обработку исходного текста. Обычно для каждой специфической задачи применяется свой словарь стоп-слов, однако для нашей задачи достаточно стандартного словаря, содержащего буквы, частицы, предлоги, союзы, местоимения, числительные. Установлено, что удаление стоп-слов из тренировочного набора значительно снижает вычислительную стоимость, а также повышает точность модели.

– К корпусу текстов применяется стемминг или лемматизация. Это позволяет сократить размер словаря и искать семантически близкие слова, а не разные формы одного слова. Стемминг – это поиск основы слова, причем не обязательно совпадающей с корнем. Он имеет высокую скорость работы, но наиболее эффективен для английского языка, так как в нем для нахождения основы слова обычно достаточно удалить окончание. Для русского языка стемминг малоэффективен, поэтому применяется более ресурсоемкий алгоритм лемматизации. Лемматизация – это процесс приведения слова к начальной форме. В данной работе лемматизация осуществлялась морфологическим анализатором MyStem [10, 11].

– Дополнение предложений до одинаковой длины с использованием нейтрального слова, так как сверточные нейронные сети способны обрабатывать только последовательности одинаковой длины.

2. Построение векторного представления слов (модель word2vec)

Как уже было сказано, на начальном этапе необходимо перевести слова естественного языка в форму, пригодную для анализа сверточной нейронной сетью. Для этого лучше всего подходит векторное представление слов. Кроме того, среди всех моделей выберем ту, которая наиболее точно отражает реальные взаимосвязи между словами, а именно семантическую близость. Отметим, что модель не должна быть слишком требовательной к вычислительным ресурсам, чтобы было возможно совершать обучение сети на достаточно больших объемах данных.

Для выявления семантических связей между словами воспользуемся предположением лингвистики – дистрибутивной гипотезой: лингвистические единицы, встречающиеся в схожих контекстах, имеют близкие значения.

Во многих моделях обработки текстов входные данные кодируются унарным кодом (one-hot encoding) – вектором, размерность которого равна мощности словаря. Элемент, соответствующий

номеру слова в словаре, равен единице, а остальные элементы равны нулю. Однако у этого метода есть ряд существенных недостатков:

– словари естественных языков могут быть достаточно объемными и исчисляться десятками и сотнями тысяч слов; следовательно, если каждое слово кодировать таким вектором, объем данных становится слишком большим;

– при таком способе кодирования теряется связь между словами: все слова считаются разными и никак не связанными между собой.

В силу вышесказанного, one-hot encoding не подходит для анализа семантической близости слов. Поэтому для данной задачи воспользуемся другим способом кодирования – распределенным представлением слов.

Распределенное (или векторное) представление слов – это способ представления слов в виде векторов евклидова пространства, размерность которого обычно равна нескольким сотням. Основная идея заключается в том, что геометрические отношения между точками евклидова пространства будут соответствовать семантическим отношениям между словами. Например, слова, представленные двумя близко расположенными точками векторного пространства, будут, скорее всего, синонимами или просто тесно связанными по смыслу словами. Семантическая близость слов вычисляется как расстояние между векторами, для чего используется так называемая косинусная мера [12].

В 2013 г. группой исследователей Google под руководством Томаша Миколова была разработана нейросетевая модель для анализа семантики естественных языков, названная word2vec. В ее основу легли идея распределенного представления слов и дистрибутивная гипотеза, позволяющая рассматривать тексты с точки зрения статистики.

Word2vec включает в себя две различные архитектуры – CBOW (Continuous Bag of Words – непрерывный мешок слов) и Skip-gram. CBOW пытается предсказать слово, исходя из текущего контекста, а Skip-gram, наоборот, пытается предсказать контекст по текущему слову. Для реализации модели была выбрана архитектура Skip-gram, которая, несмотря на меньшую скорость обучения, лучше работает с редкими словами.

Предварительно обработанный текст можно подавать на вход модели, после чего будут выполнены следующие действия:

– считывается корпус текстов и рассчитывается, сколько раз в нем встретилось каждое слово;

– из этих слов формируется словарь, который сортируется по частоте слов; также из словаря для сокращения его размера удаляются редкие слова;

– модель идет по субпредложению (обычно предложение исходного текста или абзац) окном определенного размера; под размером окна понимается максимальная длина между текущим словом и словом, которое предсказывается. Оптимальный размер окна – 10 слов;

– к данным, находящимся в текущем окне, применяется нейронная сеть прямого распространения с линейной функцией активации скрытого слоя и функцией активации softmax для выходного слоя.

Из всего вышесказанного ясно, что матрицы, задающие скрытый и выходной слои, получаются чрезвычайно большими. Это делает обучение сети долгим процессом. Поэтому используются различные оптимизации, которые позволяют существенно снизить временные и вычислительные затраты, незначительно потеряв в точности. Одной из таких модификаций является субсемплирование. Дело в том, что в больших корпусах некоторые слова могут встречаться сотни миллионов раз. Такие слова зачастую несут меньшую информационную ценность, чем редкие слова. Чтобы избежать дисбаланса между редкими и часто встречающимися словами, используется простой подход: каждое слово отбрасывается с вероятностью, зависящей от частоты вхождения этого слова в текст.

В качестве модели word2vec была выбрана реализация из библиотеки Gensim [13]. Гиперпараметры модели:

– размерность векторного пространства – 300;

– размер сканирующего окна – 10;

– константа в формуле субсемплирования – 0,00001;

– количество эпох – 5.

В качестве обучающих данных был выбрал корпус русскоязычных текстов Максима Мошкова [14]. Он содержит более 25 тыс. книг общим объемом примерно 450 млн слов. Сформированный словарь содержал около 1,3 млн слов.

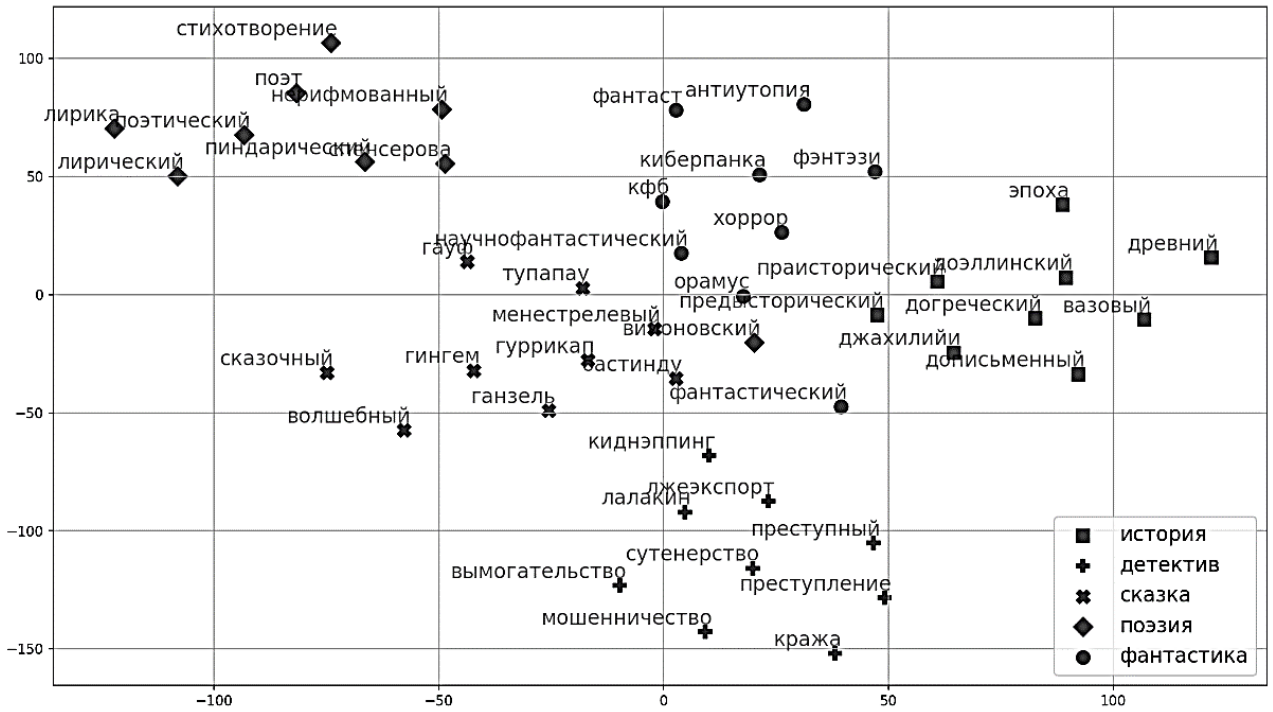

Рис. 1. Диаграмма кластеров слов различных жанров
Fig. 1. Diagram of clusters of words of various genres

Результат обучения модели можно видеть на рис. 1, где видно четкое выделение кластеров, пересекающихся лишь по тем словам, которые действительно могут быть отнесены к нескольким группам одновременно.

3. Сверточная нейронная сеть

Сверточная нейронная сеть – это специальная архитектура нейронной сети, основной принцип которой заключается в том, что обработка некоторой области данных осуществляется независимо от расположения этой области. Применительно к задачам обработки естественного языка сверточные сети позволяют анализировать семантику слов в зависимости от их контекста, так как в большинстве случаев, достаточно рассмотреть сравнительно небольшой фрагмент текста. Рассмотренный подход может быть ошибочным для некоторых слов, лексическое значение которых правильно определяется на основе литературного произведения целиком или значительной его части, однако доля таких слов в тексте, как правило, незначительна.

Входные данные представляют собой матрицу, размерности которой равны количеству предложений в обучающей выборке и максимальной длине предложений (при этом каждое слово заменено своим векторным представлением). Заметим, что при таком представлении данных имеет смысл осуществлять свертку только по одному измерению – по ширине, поэтому сверточные фильтры будут одномерными.

За основу архитектуры сети была взята конфигурация, предложенная в работе [15]. На основе анализа экспериментальных результатов были выявлены некоторые недостатки, которые были устранены следующими модификациями:

– для борьбы с переобучением был добавлен дополнительный слой Dropout (на каждом этапе обучения некоторые нейроны исключаются из рассмотрения, что в некотором смысле приводит к рассмотрению новой конфигурации сети и препятствует чрезмерной адаптации нейронов друг к другу);

– проблема внутреннего сдвига переменных (возникает при использовании мини-батчей при обучении глубоких нейронных сетей) решалась применением нормализации;

– сформирован сверточный блок с применением фильтров разных размеров, что привело к увеличению точности классификации;

– увеличено число полносвязных слоев;

– архитектура сети была доработана для осуществления классификации на произвольное количество классов: функция активации последнего слоя была заменена на softmax, что позволяет интерпретировать выход сети как вектор вероятностей принадлежности текста каждому классу.

На основе результатов многочисленных экспериментов были выбраны следующие гиперпараметры модели:

– слои Dropout: вероятности 0,5 (для входа блоки свертки) и 0,8 (для выхода блока свертки);

– слои свертки: размеры одномерных фильтров – 3, 5, 8; количество фильтров – 10; функция активации – Relu;

– слои субдискретизации: функция субдискретизации – взятие максимума;

– полносвязный слой: число нейронов 50, функция активации – Relu;

– выходной (полносвязный) слой: число нейронов равно количеству классов (в нашем случае – 5), функция активации – Softmax;

– размер одного мини-батча: 64.

Схематично разработанная архитектура представлена на рис. 2.

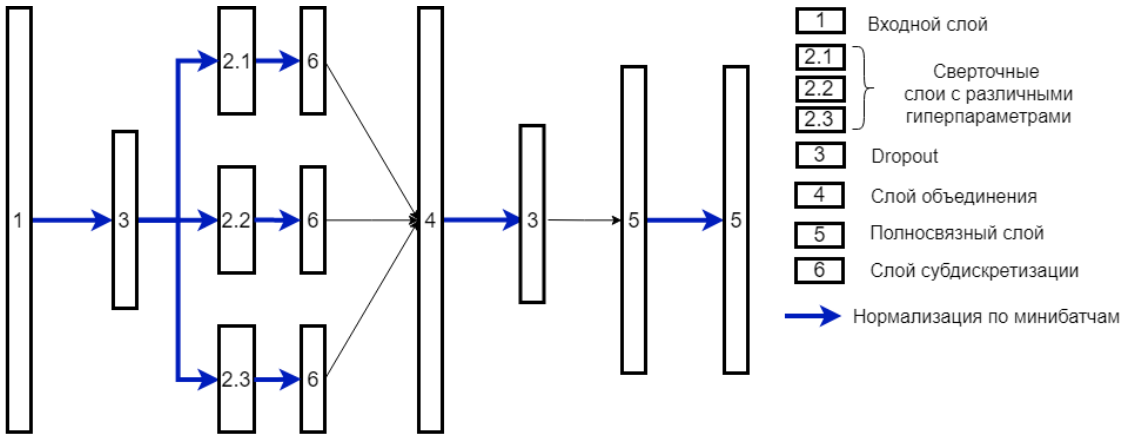

Рис. 2. Архитектура разработанной сети
Fig. 2. Architecture of the developed network

В качестве алгоритма обучения был выбран адаптивный алгоритм градиентного спуска – Adam. Он основан на следующей идее: шаг изменения должен быть меньше у тех параметров, которые в большей степени варьируют в данных, и больше у тех, которые меньше изменяются на различных примерах. Как показывает практика, такой метод обучения работает эффективнее и сходится к правильным весам быстрее, чем стохастический градиентный спуск. Несмотря на свои преимущества, адаптивные варианты градиентного спуска не решают проблему переобучения. Поэтому необходимо следить за качеством обобщающей способности модели [16].

4. Результаты обучения

Для того чтобы экспериментально проверить эффективность работы построенной модели, было выбрано пять классов: история, детективы, детская литература, поэзия и песни, фантастика и фэнте-

зи. Среди данных классов наиболее специфичным для распознавания является класс «поэзия и песни». Дело в том, что из-за применения процесса лемматизации на этапе предварительной обработки данных, текст теряет рифму и стихотворный размер. Кроме того, стихотворные произведения обычно обладают сравнительно небольшой длиной. Это затрудняет распознавание текстов данного класса на основе семантики. Как уже было сказано, на вход сети подаются дополненные до одинаковой длины предложения. Поэтому сеть может использовать информацию о количестве добавленных нейтральных слов для классификации не только по семантике, но и по длине предложений.

Сеть обучалась 100 эпох: точность на тренировочной выборке составила 78,64%, точность на тестовой выборке – более 73,12%. График зависимости ошибки на тренировочных и тестовых данных от количества эпох приведен на рис. 3.

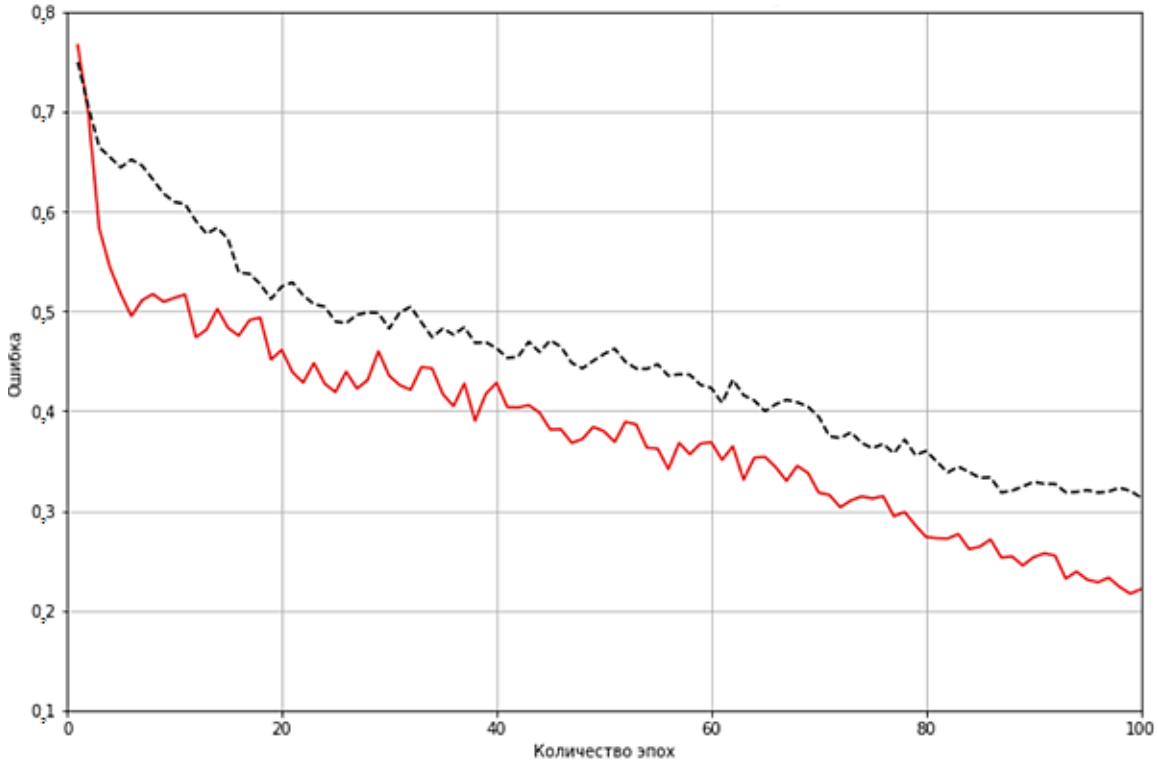

Рис. 3. Зависимость ошибки на тренировочных и тестовых данных от количества эпох
(сплошная линия – ошибка на тренировочных данных, пунктирная – ошибка на тестовых данных)
Fig. 3. Correlation of an error and the number of epochs for training and testing data
(solid line – an error on training sample, dotted line – an error on testing sample)

Как видно из графика, в течение последних 15 эпох ошибка на валидационной выборке существенно не менялась, тогда как ошибка на тестовой выборке продолжала снижаться. Чтобы предотвратить переобучение, тренировка модели была остановлена. Каждые 5 эпох производилось сохранение весов, что позволило в качестве итоговой модели выбрать сеть с минимальной ошибкой на тестовой выборке.

Задача классификации текстов по темам или жанрам решалась во многих исследованиях. Наибольший интерес представляет работа [17], в которой классификация велась по темам. В работе использовались различные модели машинного обучения, в частности сверточные (полученная точность – 70,46%) и рекуррентные (точность – 72,12%) нейронные сети, метод опорных векторов (точность – 70,22%). Следует отметить, что нами была достигнута более высокая точность классификации по сравнению с аналогичными архитектурами сверточных нейронных сетей, а также рекуррентными сетями, которые зачастую показывают наивысшие результаты при анализе текстовых последовательностей.

Рассмотрим работу сети на некоторых примерах (таблица).

Примеры работы модели

Название жанра	Произведение	Вероятность принадлежности произведения каждому классу				
		История	Детективы	Детская литература	Поэзия и песни	Фантастика и фэнтези
История	«Петр I» А.Н. Толстой	**0,4229827**	0,20211824	0,10318473	0,09724073	0,17456163
Детективы	«Собака Баскервилей» Артур Конан Дойл	0,17708729	**0,4044393**	0,13971927	0,07812358	0,20063058
Детская литература	«Малыш и Карлсон» Астрид Линдгрен	0,11668574	0,13317753	**0,38233585**	0,10727942	0,26052145
Поэзия и песни	«Руслан и Людмила» А.С. Пушкин	0,16811042	0,12634562	0,16529457	**0,37694713**	0,16330227
Поэзия и песни	«Привет, Андрей» И.Ю. Николаев	0,05886933	0,09415500	0,05023616	**0,73858399**	0,05815552
Фантастика и фэнтези	«Гарри Поттер» Дж.К. Роулинг	0,10806894	0,10925354	0,36183408	0,02905480	**0,39178870**

Видно, что вероятность принадлежности произведения определенному жанру вполне коррелирует с литературным пониманием этого текста. Действительно, поэма «Руслан и Людмила», в первую очередь рассматривается как стихотворное произведение, однако содержит элементы исторического рассказа и фэнтези. «Собака Баскервилей», например, со значительной вероятностью относится к классу «Фантастика и фэнтези», что в некоторой степени объясняется высокой степенью мистицизма в данном произведении. Вероятность того, что «Гарри Поттер» относится к классу «Фантастика и фэнтези», близка к вероятности класса «Детская литература», что также соответствует нашим представлениям.

Заключение

Таким образом, для решения поставленной задачи была разработана архитектура сверточной нейронной сети, на вход которой подавались векторные представления слов, полученных на основе модели word2vec. Предложенная модель сверточной нейронной сети является корректной и достаточно точно отражает литературные представления о жанре текстов. Поэтому данная модель может быть применена для автоматизации обработки текстов в корпусной лингвистике.

ЛИТЕРАТУРА

1. Батраева И.А., Крючкова А.А. Разработка программного обеспечения диалектологических корпусов // Компьютерные науки и информационные технологии : материалы Междунар. науч. конф. Саратов : Наука, 2018. С. 45–49.
2. Bai S., Kolter J.Z., Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. 2018. arXiv preprint arXiv: 1803.01271.
3. Conneau A., Schwenk H., Barrault L., LeCun Y. Very deep convolutional networks for text classification. 2017. arXiv preprint arXiv: 1606.01781.
4. Zhang X., Zhao J., LeCun Y. Character-level Convolutional Networks for Text Classification. 2016. arXiv preprint arXiv: 1509.01626.
5. Yin W., Kann K., Yu M., Schütze H. Comparative study of CNN and RNN for natural language processing. 2017. arXiv preprint arXiv: 1702.01923.
6. Yogatama D., Dyer Chr., Ling W., Blunsom Ph. Generative and discriminative text classification with recurrent neural networks. 2017. arXiv preprint arXiv: 1703.01898.
7. Rong Xin. Word2vec parameter learning explained. 2014. arXiv preprint arXiv: 1411.2738.
8. NLTK 3.4 documentation. URL: http://www.nltk.org/ (accessed: 08.02.2019).
9. Train NLTK punkt tokenizers. URL: https://github.com/mhq/train_punkt (accessed: 08.02.2019).
10. Яндекс технология MyStem. URL: https://tech.yandex.ru/mystem/ (дата обращения: 08.02.2019).
11. Segalovich I. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine // Proc. of the International Conf. on Machine Learning. Models, Technologies and Applications. MLMTA'03, June 23–26, 2003, Las Vegas, Nevada, USA. P. 1–8.
12. Mikolov T., Sutskever I., Chen K., Corrado G., Dean J. Distributed representations of words and phrases and their compositionality. 2013. arXiv preprint arXiv: 1310.4546

13. Gensim documentation. URL: https://radimrehurek.com/gensim/tutorial.html (accessed: 10.02.2019).
14. Библиотека Максима Мошкова. URL: http://lib.ru (дата обращения: 20.01.2019).
15. Kim Y. Convolutional neural networks for sentence classification // Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP 2014). 2014. P. 1746–1751.
16. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, который извлекают знания из данных. М. : ДМК Пресс, 2015. 402 с.
17. Kamran K., Donald E., Mojtaba H., Kiana J.M., Matthew S., Laura E. HDLTex: Hierarchical Deep Learning for Text Classification. 2017. arXiv preprint arXiv:1709.08267.

Поступила в редакцию 2 июня 2019 г.

Batraeva I.A., Nartsev A.D., Lezgyan A.S. (2020) USING THE ANALYSIS OF SEMANTIC PROXIMITY OF WORDS IN SOLVING THE PROBLEM OF DETERMINING THE GENRE OF TEXTS WITHIN DEEP LEARNING. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitelnaja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 50. pp. 14–22

DOI: 10.17223/19988605/50/2

The relevant objective in the processing of text corpora is the classification of texts by topics and genres. Usually this work is done manually, so processing large text corpora is an extremely long process. Moreover, an unambiguous classification is not always possible: in most cases, the same text can be attributed to several topics and genres, with only one of them being the principal one. Therefore, the full automation of the classification process or limiting the choice of a researcher to the list of the most likely topics and genres is of practical interest.

To solve the problem, the authors propose to use convolutional neural networks, which, on the one hand, are efficient in classifications, and, on the other hand, are not used and studied properly for text recognition.

To present the data in a form suitable for processing by a convolutional neural network, the word2vec model was chosen. This model allows us to conduct vector representations of words that reflect their semantic proximity. To implement the word2vec model, the Skip-gram architecture was chosen, which, despite the slow learning rate, works well with rare words.

Based on the results of numerous experiments, the most optimal model hyperparameters were selected. The output of a trained model is the probability of attribution of a work to each class. Based on the analysis of the obtained results, we can conclude that the proposed model of the convolutional neural network is correct and fairly accurately reflects the literary perception of the genre.

Keywords: machine learning; convolutional neural networks; word2vec model; text natural language processing.

BATRAEVA Inna Aleksandrovna (Candidate of Physics and Mathematics, Head of the Department of Programming Technologies, Saratov State University, Saratov, Russian Federation).
E-mail: BatraevaIA@info.sgu.ru

NARTSEV Andrey Dmitrievich (Saratov State University, Saratov, Russian Federation).
E-mail: narcev.andrey@gmail.com

LEZGYAN Artem Sarkisovich (Saratov State University, Saratov, Russian Federation).
E-mail: lezgyan@yandex.ru

REFERENCES

1. Batraeva, I.A. & Kryuchkova, A.A. (2018) Razrabotka programmnogo obespecheniya dialektologicheskikh korpusov [Developing software for dialect corpora]. In: Tverdokhlebov, V. (ed.) *Komp'yuternye nauki i informatsionnye tekhnologii* [Computer Science and Information Technologies]. Saratov: Saratov State University. pp. 45–49.
2. Bai, S., Kolter, J.Z. & Koltun, V. (2018) *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling*. arXiv preprint arXiv: 1803.01271
3. Conneau, A., Schwenk, H., Barrault, L. & LeCun, Y. (2017) *Very deep convolutional networks for text classification*. arXiv preprint arXiv: 1606.01781
4. Zhang, X., Zhao, J. & LeCun, Y. (2016) *Character-level Convolutional Networks for Text Classification*. arXiv preprint arXiv: 1509.01626
5. Yin, W., Kann, K., Yu, M. & Schütze, H. (2017) *Comparative study of CNN and RNN for natural language processing*. arXiv preprint arXiv: 1702.01923
6. Yogatama, D., Dyer, Chr., Ling, W. & Blunsom, Ph. (2017) *Generative and discriminative text classification with recurrent neural networks*. arXiv preprint arXiv: 1703.01898
7. Rong, Xin. (2014) *Word2vec parameter learning explained*. arXiv preprint arXiv: 1411.2738
8. NLTK.org. (n.d.) *NLTK 3.4 documentation*. [Online] Available from: http://www.nltk.org/ (Accessed: 8th April 2019).

9. Github.com. (n.d.) *Train NLTK punkt tokenizers*. [Online] Available from: https://github.com/mhq/train_punkt (Accessed: 8th April 2019).
10. Yandex.ru. (n.d.) *Yandeks tekhnologiya MyStem* [MyStem Yandex Technology]. [Online] Available from: https://tech.yandex.ru/mystem/ (Accessed: 8th April 2019).
11. Segalovich, I. (2003) A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. *Models, Technologies and Applications. MLMTA'03*. Proc. of the International Confeerence on Machine Learning. June 23–26, 2003. Las Vegas, Nevada, USA. pp. 1–8.
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013) *Distributed representations of words and phrases and their compositionality*. arXiv preprint arXiv: 1310.4546
13. Radimrehurek.com. (n.d.) *Gensim documentation*. [Online] Available from: https://radimrehurek.com/gensim/tutorial.html (Accessed: 8th April 2019).
14. Lib.ru. (n.d.) *Biblioteka Maksima Moshkova* [Maxim Moshkov's Library]. [Online] Available from: http://lib.ru (Accessed: 8th April 2017).
15. Kim, Y. (2014) Convolutional neural networks for sentence classification. *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP 2014)*. pp. 1746–1751.
16. Flach, P. (2015) *Mashinnoe obuchenie. Nauka i iskusstvo postroeniya algoritmov, kotoryy izvlekayut znaniya iz dannykh* [Machine Learning: The Art and Science of Algorithms That Make Sense of Data]. Translated from English. Moscow: DMK Press.
17. Kamran, K., Donald, E., Mojtaba, H., Kiana, J., Matthew, S. & Laura, E. (2017) *HDLTex: Hierarchical Deep Learning for Text Classification*. arXiv preprint arXiv:1709.08267