DOI: 10.18137/RNU.V9187.22.02.P.129 УДК 004.85

Д.С. Бузин, М.Т. Азизов

АЛГОРИТМЫ МАШИННОГО ОБУЧЕНИЯ ДЛЯ АНАЛИЗА ТОНАЛЬНОСТИ ВЫСКАЗЫВАНИЙ

Аннотация. Проводится комплексный анализ технологии машинного обучения. В ходе исследования разработаны алгоритмы глубокого обучения для анализа тональности текста и проведено сравнение их эффективности с другими классификаторами на основе алгоритмов машинного обучения.

Ключевые слова: машинное обучение, тональность, нейросеть.

D.S. Buzin, M.T. Azizov

MACHINE LEARNING ALGORITHMS FOR SENTIMENT ANALYSIS

Abstract. The study provides a comprehensive analysis of machine learning technology. In the course of the study, deep learning algorithms were developed for analyzing the sentiment of the text and a comparison was made of their effectiveness with other classifiers based on machine learning algorithms.

Keywords: machine learning, tonality, neural network.

Введение

В последнее десятилетие в интернете генерируются огромные объемы данных, которые несут в себе множество информации, в том числе потребительское мнение, политические настроения, экстремистские взгляды и высказывания. Такие данные, как правило, являются неструктурированным текстом. Чтобы стало возможным использование данной информации, необходимо ее классифицировать и систематизировать. Одной из самых сложных проблем классификации является анализ тональности текста.

Тональность – это высказывание мнения автора об объекте, событии, процессе и их свойствах, выраженное в эмоциональной оценке. Анализ тональности можно рассматривать как процесс классификации, целью которого является присвоение текстам некоторой категории из конкретного набора. Наиболее простой считается классификация в одномерном эмотивном пространстве, то есть в пространстве двух тональностей – позитивной или негативной [4].

Анализ тональности позволяет упростить получение обратной связи о продуктах и услугах, сделать проведение анализа рекламной и PR-деятельности эффективнее; кроме того, он играет важнейшую роль в принятии обоснованных решений о маркетинговых стратегиях. Методы анализа тональности помогают в решении проблемы обеспечения безопасности пользователей в интернете, измерении настроений общества, что важно как для политики, так и для рыночного прогнозирования.

Обзор предметной области и архитектура глубоких нейронных сетей

Анализ тональности текста представляет собой набор алгоритмов компьютерной лингвистики, основной целью которых является извлечение из высказываний мнений авторов по отношению к объектам, речь о которых идет в тексте.

Часто в литературе можно встретить использование значения нейтральной тональности, под которой подразумевается отсутствие эмоциональной окраски в тексте [3].

Бузин Дмитрий Сергеевич
магистрант кафедры информационной безопасности. МИРЭА – Российский технологический университет, Москва. Сфера научных интересов: информационная безопасность; информатика и вычислительная техника.
Электронный адрес: Buzin.97@mail.ru

Азизов Мукум Тимурович
магистрант кафедры информационной безопасности. МИРЭА – Российский технологический университет, Москва. Сфера научных интересов: информационная безопасность; информатика и вычислительная техника.
Электронный адрес: mukimazizov@icloud.com

В исследовании для решения поставленной задачи рассмотрены две различные архитектуры нейронных сетей – рекуррентная и сверточная. Это связано с тем, что решения на основе нейронных сетей показывают лучшие результаты в самых различных областях человеческого знания, в том числе в области анализа тональности текста.

Convolutional neural network (CNN). CNN первоначально были разработаны для обработки изображений, однако они успешно справляются с решением задач в сфере автоматической обработки текстов [11]. Основная идея сверточной нейронной сети – постепенный переход от конкретных особенностей текста к абстрактным вплоть до получения высокоуровневых понятий. При этом сеть сама конфигурируется, выделяя иерархию существенных абстрактных деталей и фильтруя маловажные.

Архитектура сверточной нейронной сети представлена на Рисунке 1.

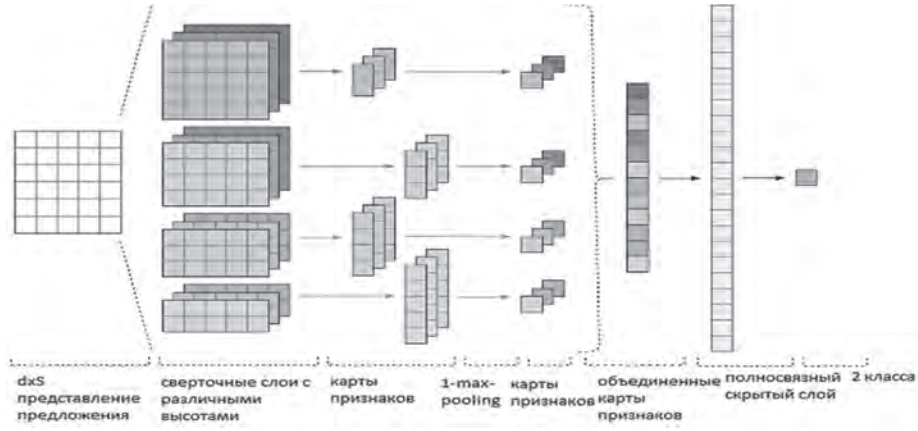

Рисунок 1. Архитектура сверточной нейронной сети

Recurrent neural network (RNN). Главное преимущество рекуррентных нейронных сетей – использование предыдущего состояния сети для получение текущего, то есть реализация памяти.

Данный класс сетей эффективно используется для решения задач анализа тональности текста, поскольку благодаря наличию обратных связей позволяет анализировать последовательности данных, в которых важно, в каком порядке идут значения.

Главное отличие рекуррентных сетей друг от друга заключается в том, как обрабатывается ячейка памяти внутри них. Связанные по смыслу слова могут стоять в тексте на достаточно большом расстоянии. Таким образом, разрыв между актуальной информацией и точкой ее применения может стать очень большим. По мере роста этого расстояния RNN теряют способность связывать информацию. Для запоминания информации на долгие периоды времени была разработана долгая краткосрочная память (Long Short Term Memory (LSTM)) [10].

LSTM – особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям.

Архитектура LSTM-блока представлена на Рисунке 2.

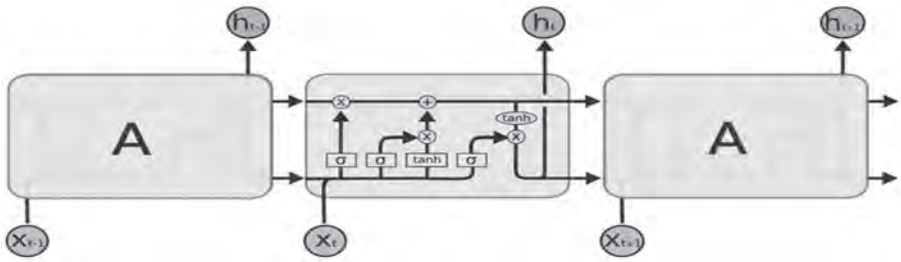

Рисунок 2. Развернутая по времени архитектура LSTM

Сначала определяется, от какой информации в ячейке памяти можно избавиться. Для этого на первом слое вычисляются множители к компонентам вектора памяти:

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right), \tag{1}$$

где h_{t-1} – значение на выходе предыдущей итерации; x_t – сигнал на входе в момент времени t.

Далее вычисляется новая информация, которая записывается в ячейку памяти, или наблюдение C^*:

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right); \tag{2}$$

$$C^* = tahn\left(\sigma\left(W_C\left[h_{t-1}, x_t\right] + b_C\right)\right). \tag{3}$$

На третьем слое получают новое состояние ячейки памяти C_t как линейную комбинацию памяти и наблюдения:

$$C_t = f_t C_{t-1} + i_t C. \tag{4}$$

На последнем шаге вычисляется значение выходного нейрона σ_t:

$$\sigma_t = \sigma\left(W_0\left[h_{t-1}, x_t\right] + b_0\right); \tag{5}$$

$$h_t = \sigma_t tah(C_t). \tag{6}$$

Полученные значения h_t и C_t поступают на вход сети в момент времени $t + 1$. Обучение сети реализуется на основе алгоритма обратного распространения ошибки [6].

Описание процесса обработки информации и архитектура реализованных глубоких нейронных сетей

В качестве основного языка разработки выбран Python, который обладает большим количеством библиотек для реализации алгоритмов анализа тональности текста.

В ходе выполнения данной работы реализованы такие классические классификаторы, как наивный байесовский, метод опорных векторов, случайный лес решений, а также ре-

куррентная и сверточная нейронные сети. Текст классифицировался на два класса – положительный и отрицательный.

Реализацию можно разбить на 2 модуля – предварительная обработка данных и реализация алгоритма.

Данные были взяты из корпуса коротких текстов Юлии Рубцовой, сформированных на основе русскоязычных сообщений из Twitter [2]. Он содержит 114 991 положительных, 111 923 отрицательных твитов, а также базу неразмеченных твитов объемом 17 639 674 записей, выложенных с конца ноября 2013 года до конца февраля 2014 года.

Как видно из Рисунка 3, каждый текст в базе данных имеет множество атрибутов: дату публикации, имя автора, количество ретвитов, количество друзей пользователя и др. В данной работе было использовано два атрибута – текст твита (ttext) и класс, к которому он принадлежит, – положительный или отрицательный (ttype).

Рисунок 3. Корпус коротких текстов Юлии Рубцовой

Для подготовки данных к классификации или обучению необходимо провести их предварительную обработку, поскольку данные могут быть зашумлены или обладать символами, не несущими никакой смысловой нагрузки.

Этапы предварительной обработки данных:

- преобразование всего текста в нижний регистр;
- замена всех знаков пунктуации пробелами;
- разбиение текста на слова по пробелам;
- удаление всех ссылок;
- стеммизация;
- замена одинаковых слов на одинаковые числовые коды [9].

В итоговый словарь было добавлено 100 000 самых употребляемых слов. Записи были представлены в виде последовательностей чисел.

Процесс предварительной обработки данных представлен на Рисунке 4.

Каждый текст был отображен в массив идентификаторов токенов. Размерность вектора текста $s = 34$, поскольку при данном значении полностью покрываются все тексты в сформированном корпусе (см. Рисунок 5).

Если при распознавании количество слов в сообщении будет превышать высоту матрицы, оставшиеся слова будут отброшены, и не будут учитываться в классификации.

Кроме того, для автоматизации подбора параметров классификаторов был использован класс Grid Search CV библиотеки scikit-learn [5].

Рисунок 4. Процесс предварительной обработки данных

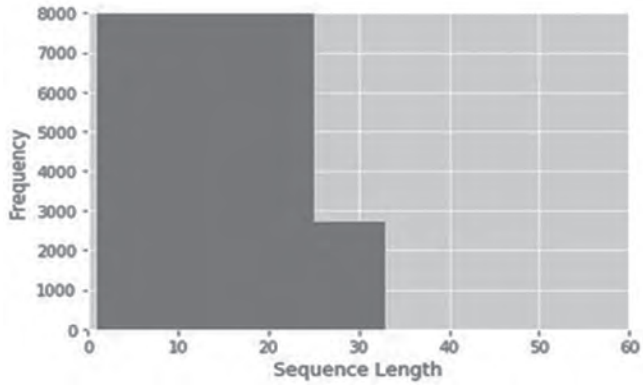

Рисунок 5. Распределение длины текстов

Реализация рекуррентной нейронной сети с LSTM-блоками. Предлагаемая архитектура сети

Для реализации нейронных сетей была использована библиотека Thensor Flow. Предлагаемая архитектура рекуррентной нейронной сети представлена на Рисунке 6.

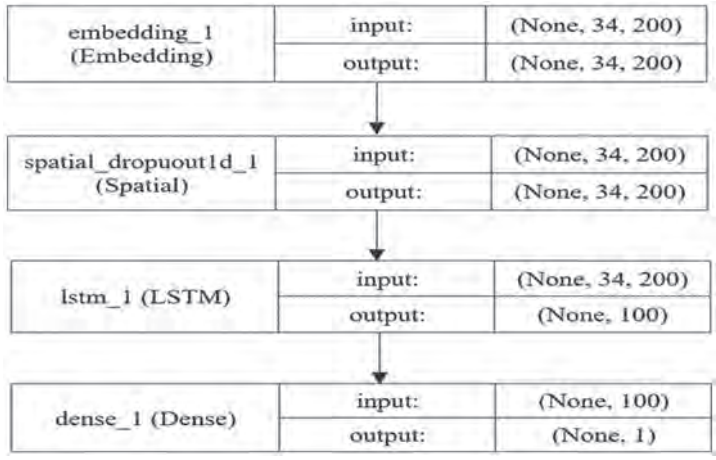

Рисунок 6. Архитектура рекуррентной сети

Реализация сверточной нейронной сети. Предлагаемая архитектура

Реализованная архитектура CNN сети представлена на Рисунке 7.

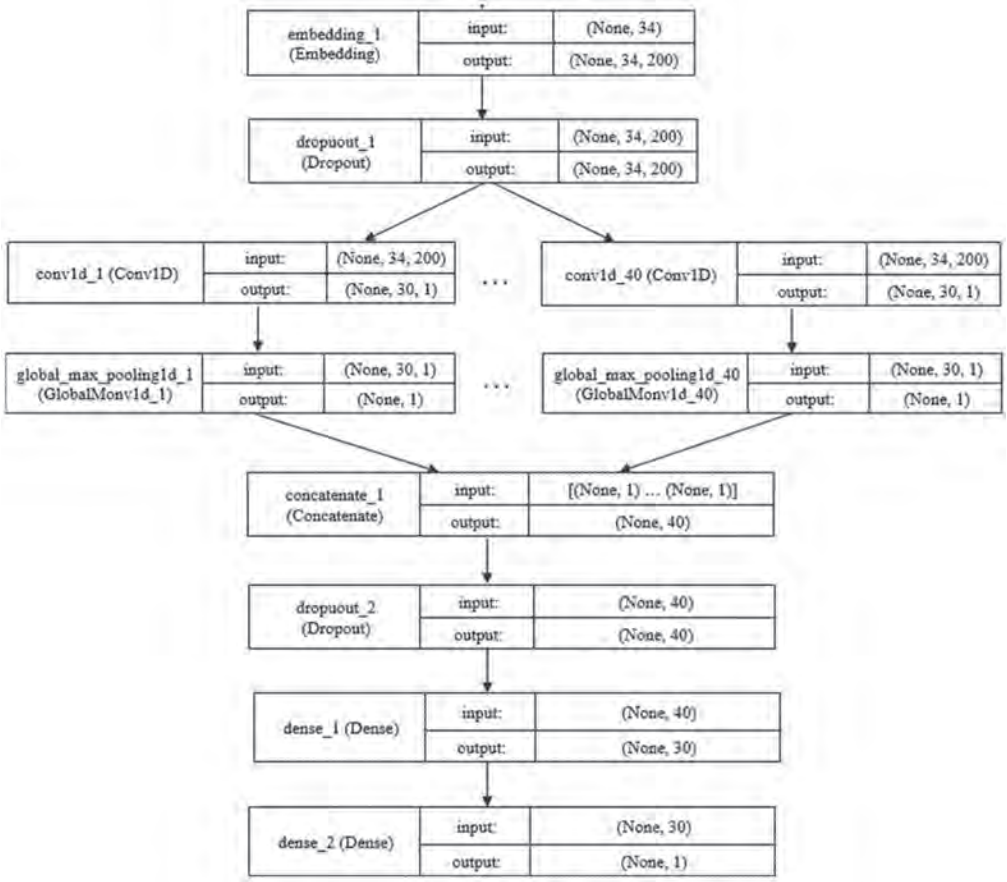

Рисунок 7. Архитектура сверточной нейронной сети

Для оценки качества классификаторов было принято решение использовать следующие характеристики [7]:

• точность (precision), которая характеризует, сколько полученных от классификатора положительных ответов являются правильными;

• полнота (recall), которая определяет, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм. Recall демонстрирует способность алгоритма обнаруживать данный класс вообще, а precision – способность отличать этот класс от других классов;

• мера F_1, которая является средним гармоническим меры точности и меры полноты. Характеризует пороговое качество классификатора. *F*-мера достигает максимума, если полнота и точность равны единице, и близка к нулю, если один из аргументов близок к нулю;

• носитель меры (support), который представляет собой количество данных каждого из классов.

Программная реализация и результаты экспериментальных исследований

В данном разделе представлены результаты работы наивного байесовского классификатора, метода опорных векторов, случайного леса решений [8], рекуррентной нейронной сети и сверточной нейронной сети [11] (см. Таблицы 1–5).

Таблица 1

Наивный байесовский классификатор

Класс	**Мера точности**	**Мера полноты**	**Мера F_1**	**Носитель меры**
0	0,7281	0,8074	0,7657	22236
1	0,7871	0,7025	0,7424	22534
total	0,7578	0,7546	0,7540	44770
Эффективность75 %			Время обучения 14,5 мин	

Таблица 2

Метод опорных векторов

Класс	**Мера точности**	**Мера полноты**	**Мера F_1**	**Носитель меры**
0	0,6269	0,5183	0,5675	22236
1	0.5941	0,6956	0,6408	22534
total	0,6104	0,6075	0,6044	44770
Эффективность 60 %			Время обучения 15 мин	

Таблица 3

Случайный лес

Класс	**Мера точности**	**Мера полноты**	**Мера F_1**	**Носитель меры**
0	0,6204	0,7167	0,6650	22236
1	0,6698	0,5672	0,6200	22534
total	0,6453	0,6415	0,6423	44770
Эффективность 64 %			Время обучения 7 мин	

Таблица 4

Рекуррентная нейронная сеть

Класс	**Мера точности**	**Мера полноты**	**Мера F_1**	**Носитель меры**
0	0,774	0,803	0,788	22236
1	0,798	0,768	0,62	22534
total	0,786	0,786	0,786	44770
Эффективность 79 %			Время обучения 57 мин	

Таблица 5

Сверточная нейронная сеть

Класс	Мера точности	Мера полноты	Мера F_1	Носитель меры
0	0,752	0,812	0,781	22236
1	0,799	0,736	0,766	22534
total	0,786	0,774	0,775	44770
Эффективность 78 %			Время обучения 7 ч	

Сравнение и обсуждение результатов

Как видно из сводной таблицы результатов (см. Таблицу 6), лучшие результаты среди классических классификаторов показал мультиномиальный байесовский классификатор, достигнув меры $F_1 = 0{,}75$.

Таблица 6

Сводная таблица результатов

Классификатор	Мера F_1	Время обучения
Наивный байесовский классификатор	0,75	14,5 мин
Методопорныхвекторов	0,60	15мин
Случайныйлес	0,64	7мин
Рекуррентнаянейроннаясеть	0,79	57мин
Сверточнаянейроннаясеть	0,78	7ч

При использовании сверточных нейронных сетей с моделью Word2Vec удалось получить результат 78 %.

Самой эффективной архитектурой для анализа тональности текста оказалась рекуррентная нейронная сеть с LSTM-блоками. Ее показатель качества составил 79 %, при этом скорость обучения значительно превысила скорость обучения CNN.

Полученные результаты показывают более высокую эффективность работы глубоких нейронных сетей по сравнению с классическими алгоритмами для анализа тональности текста.

Для наглядной демонстрации работы нейронных сетей была поставлена задача написания приложения, проводящего бинарную классификацию данных из Twitter по введенному пользователем хештегу.

Вывод представлен на Рисунке 8 [1].

Так, например, для хештега #капитанмарвел было получено 72,5 % положительных и 27,5 % отрицательных отзывов. Так как в обучающих данных не было нейтральной тональности, то положительный отзыв считался с вероятностью от 0,65, отрицательный – до 0,45, а промежуток между ними считался нейтральным. В целом видно, что людям фильм понравился.

Выводы

В ходе исследования разработаны алгоритмы глубокого обучения для анализа тональности текста и проведено сравнение их эффективности с другими классификаторами на основе алгоритмов машинного обучения.

Рисунок 8. Страница answer.html

Результаты исследования показывают, что использование глубоких нейронных сетей значительно улучшает точность анализа тональности текста. Мера F_1 классификатора на основе сверточной нейронной сети оказалась 78 %. Самую высокую эффективность показал классификатор на основе рекуррентной сети с LSTM-блоками – 79 %.

Разработано приложение, позволяющее классифицировать данные, полученные из Twitter, по хештегу на положительную и отрицательную тональность и выводить диаграмму результата.

Возможным направлением для дальнейшей работы является реализация функционала уведомления об ошибке классификации, что позволило бы провести сбор данных, на которых сеть допускает ошибки, и улучшить результаты сети путем дообучения ее на конкретных данных.

Литература

1. *Клековкина М.В., Котельников Е.В.* Метод автоматической классификации текстов по тональности, основанный на словаре эмоциональной лексики // Труды 14-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы итехнологии, электронные коллекции» – RCDL-2012 (Переславль-Залесский, 15–18 октября 2012 г.) / Вятский государственный гуманитарный университет, 2012. С. 118–123.

2. *Рубцова Ю.В.* Построение корпуса текстов для настройки тонового классификатора // Программные продукты и системы. 2015. № 1 (109). С. 72–78.

3. *Сарбасова А.Н.* Исследование методов сентимент-анализа русскоязычных текстов // Молодой ученый. 2015. № 8. С. 143–146.

4. *Стригулин К.А., Журавлева Л.В.* Анализ тональности высказываний в Twitter // Молодой ученый. 2016. № 12. С. 185–189.

5. *Chin-Sheng Yang, Hsiao-Ping Shih* (2012) A Rule-Based Approach For Effective Sentiment Analysis. PACIS. Available at: https://aisel.aisnet.org/pacis2012/181 (date of the application: 23.04.2022).

6. *Diederik P. Kingma, Jimmy Ba. Adam* (2014) A Method for Stochastic Optimization – 2014. Available at: https://arxiv.org/abs/1412.6980 (date of the application: 23.04.2022).

7. *Kang Hanhoon, YooSeong Joon, Han Dongil* (2012) Senti-lexicon and improved Naıve Bayes algorithms for sentiment analysis of restaurant reviews. Expert Systems with Applications, No. 39, pp. 6000–6010.

8. *Kaufmann J.M. J. Max Align* (2012) Maximum Entropy Parallel Sentence Alignment Tool. Mumbai: The COLING 2012 Organizing Committee, pp. 277–288.
9. *Ko Youngjoong, Seo Jungyun* (2000) Automatic text categorization by unsupervised learning. COLING-00, the 18th international conference on computational linguistics, No. 1, pp. 453–459.
10. *Li Y., Jain A.* (1998) Classification of text documents. The Computer Journal, No. 41, pp. 537–546.
11. *Li Yung-Ming, Li Tsung-Ying* (2013) Deriving market intelligence from microblogs. Decision Support Systems, No. 55, pp. 206–217.
12. *Ortigosa-Hernandez Jonathan, Rodrıguez Juan Diego, Alzate Leandro, Lucania Manuel, InzaInaki, Lozano Jose* (2012) Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. Neurocomputing, No. 92, pp. 98–115.
13. *Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, Christopher Potts* (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. EMNLP, pp. 1631–1642.
14. *Walaa Medhat, Ahmed Hassan, HodaKorashy* (2014) Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, No. 5, pp. 1093–1113.
15. *Zhang Y., Wallace B.* (2015) A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. Available at: https://arxiv.org/abs/1510.03820 (date of the application: 23.04.2022).

References

1. Klekovkina M.V., Kotel'nikov E.V. (2012) *Metod avtomaticheskо jklassifikacii tekstov po tonal'nosti, osnovannyj na slovare emocional'noj leksiki* [A method for automatic classification of texts by tonality based on a dictionary of emotional vocabulary], pp. 118–123 (in Russian).
2. Rubcova Yu.V. (2015) *Postroenie korpusa tekstov dlya nastrojki tonovogo klassifikatora* [Building a corpus of texts to set up a tone classifier]. *Programmnye produkty i sistemy*, No. 1 (109), pp. 72–78 (in Russian).
3. Sarbasova A.N. (2015) *Issledovanie metodov sentiment-analiza russkoyazychnyh tekstov* [Study of methods of sentiment analysis of Russian-language texts]. *Molodoj uchenyj*, No. 8, pp. 143–146 (in Russian).
4. Strigulin K.A., Zyuravleva L.V. (2016) *Analiz tonal'nosti vyskazyvanij v Twitter* [Sentiment Analysis on Twitter]. *Molodoj uchenyj*, No. 12, pp. 185–189 (in Russian).
5. Chin-Sheng Yang, Hsiao-Ping Shih (2012) A Rule-Based Approach For Effective Sentiment Analysis. PACIS. Available at: https://aisel.aisnet.org/pacis2012/181 (date of the application: 23.04.2022).
6. Diederik P. Kingma, Jimmy Ba. Adam (2014) A Method for Stochastic Optimization – 2014. Available at: https://arxiv.org/abs/1412.6980 (date of the application: 23.04.2022).
7. Kang Hanhoon, YooSeong Joon, Han Dongil (2012) Senti-lexicon and improved Naıve Bayes algorithms for sentiment analysis of restaurant reviews. Expert Systems with Applications, No. 39, pp. 6000–6010.
8. Kaufmann J.M. J. Max Align (2012) Maximum Entropy Parallel Sentence Alignment Tool. Mumbai: The COLING 2012 Organizing Committee, pp. 277–288.
9. Ko Youngjoong, Seo Jungyun (2000) Automatic text categorization by unsupervised learning. COLING-00, the 18th international conference on computational linguistics, No. 1, pp. 453–459.
10. Li Y., Jain A. (1998) Classification of text documents. The Computer Journal, No. 41, pp. 537–546.
11. Li Yung-Ming, Li Tsung-Ying (2013) Deriving market intelligence from microblogs. Decision Support Systems, No. 55, pp. 206–217.

12. Ortigosa-Hernandez Jonathan, Rodrıguez Juan Diego, Alzate Leandro, Lucania Manuel, InzaInaki, Lozano Jose (2012) Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. Neurocomputing, No. 92, pp. 98–115.
13. Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, Christopher Potts (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. EMNLP, pp. 1631–1642.
14. Walaa Medhat, Ahmed Hassan, Hoda Korashy (2014) Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, No. 5, pp. 1093–1113.
15. Zhang Y., Wallace B. (2015) A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. Available at: https://arxiv.org/abs/1510.03820 (date of the application: 23.04.2022).