

<p>Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics</p>	<p>На пути к структуре обработки в реальном времени, основанной на улучшенных распределенных рекуррентных вариантах нейронной сети с fastText для анализа больших данных в социальных сетях</p>
<p>Badr Ait Hammou*,a, Ayoub Ait Lahcena,b, Salma Moulinea a <i>LRIT, Associated Unit to CNRST (URAC 29), Rabat IT Center, Faculty of Sciences, Mohammed V University, Rabat, Morocco</i> b <i>LGS, National School of Applied Sciences (ENSA), Ibn Tofail University, Kenitra, Morocco</i></p>	
<p>ABSTRACT</p>	<p>Аннотация</p>
<p>Big data generated by social media stands for a valuable source of information, which offers an excellent opportunity to mine valuable insights. Particularly, User-generated contents such as reviews, recommendations, and users' behavior data are useful for supporting several marketing activities of many companies. Knowing what users are saying about the products they bought or the services they used through reviews in social media represents a key factor for making decisions. Sentiment analysis is one of the fundamental tasks in Natural Language Processing. Although deep learning for sentiment analysis has achieved great success and allowed several firms to analyze and extract relevant information from their textual data, but as the volume of data grows, a model that runs in a traditional environment cannot be effective, which implies the importance of efficient distributed deep learning models for social Big Data analytics. Besides, it is known that social media analysis is a complex process, which involves a set of complex tasks. Therefore, it is important to address the challenges and issues of social big data analytics and enhance the performance of deep learning techniques in terms of classification accuracy to obtain better decisions.</p>	<p>Большие данные, генерируемые социальными сетями, представляют собой ценный источник информации, который предоставляет отличную возможность для получения ценных знаний. В частности, пользовательский контент, такой как обзоры, рекомендации и данные о поведении пользователей, полезен для поддержки нескольких маркетинговых мероприятий многих компаний. Знание того, что пользователи говорят о продуктах, которые они купили, или услугах, которые они использовали в обзорах в социальных сетях, является ключевым фактором для принятия решений. Анализ настроений - одна из фундаментальных задач обработки естественного языка. Хотя глубокое изучение анализа настроений достигло больших успехов и позволило нескольким фирмам анализировать и извлекать соответствующую информацию из своих текстовых данных, но по мере роста объема данных модель, работающая в традиционной среде, не может быть эффективной, что подразумевает важность эффективные распределенные модели глубокого обучения для социальной аналитики больших данных. Кроме того, известно, что анализ социальных сетей - это сложный процесс, который включает в себя набор сложных задач. Поэтому важно решать задачи и проблемы анализа больших данных в социальных сетях и повышать производительность методов глубокого обучения с точки зрения точности классификации для получения более эффективных решений.</p>
<p>In this paper, we propose an approach for sentiment analysis, which is devoted to adopting fastText with Recurrent neural network variants to represent textual data efficiently. Then, it employs the new representations to perform the classification</p>	<p>В этой статье мы предлагаем подход к анализу настроений, который посвящен принятию fastText с рекуррентными вариантами нейронной сети для эффективного представления</p>

task. Its main objective is to enhance the performance of well-known Recurrent Neural Network (RNN) variants in terms of classification accuracy and handle large scale data. In addition, we propose a distributed intelligent system for real-time social big data analytics. It is designed to ingest, store, process, index, and visualize the huge amount of information in real-time. The proposed system adopts distributed machine learning with our proposed method for enhancing decision-making processes. Extensive experiments conducted on two benchmark data sets demonstrate that our proposal for sentiment analysis outperforms well-known distributed recurrent neural network variants (i.e., Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), and Gated Recurrent Unit (GRU)). Specifically, we tested the efficiency of our approach using the three different deep learning models. The results show that our proposed approach is able to enhance the performance of the three models. The current work can provide several benefits for researchers and practitioners who want to collect, handle, analyze and visualize several sources of information in real-time. Also, it can contribute to a better understanding of public opinion and user behaviors using our proposed system with the improved

текстовых данных. Затем он использует новые представления для выполнения задачи классификации. Его главная цель - повысить производительность хорошо известных вариантов рекуррентной нейронной сети (RNN) с точки зрения точности классификации и обработки крупномасштабных данных. Кроме того, мы предлагаем распределенную интеллектуальную систему для анализа социальных данных в реальном времени. Он предназначен для приема, хранения, обработки, индексации и визуализации огромного объема информации в режиме реального времени. Предлагаемая система использует распределенное машинное обучение с нашим предложенным методом для улучшения процессов принятия решений. Обширные эксперименты, проведенные с двумя наборами эталонных данных, демонстрируют, что наше предложение по анализу настроений превосходит хорошо известные варианты распределенной рекуррентной нейронной сети (то есть длинную кратковременную память (LSTM), двунаправленную долгосрочную кратковременную память (BiLSTM) и управляемый рекуррентный блок (GRU)). В частности, мы проверили эффективность нашего подхода с использованием трех различных моделей глубокого обучения. Результаты показывают, что предлагаемый нами подход способен повысить производительность трех моделей. Текущая работа может предоставить ряд преимуществ для исследователей и практиков, которые хотят собирать, обрабатывать, анализировать и визуализировать несколько источников информации в режиме реального времени. Кроме того, это может способствовать лучшему пониманию общественного мнения и поведения пользователей с использованием нашей предлагаемой системы с улучшенными

1. Introduction

In the era of Big Data, the world's largest technology organizations like Microsoft, Amazon, and Google have collected massive amounts of data estimated at the size of exabytes or larger. Social media companies like Twitter, Facebook, and YouTube have billions of users. Therefore, many businesses are employing social media to keep in touch with their clients, and promote the

1. Введение

В эпоху Больших Данных крупнейшие в мире технологические организации, такие как Microsoft, Amazon и Google, собрали огромные объемы данных, оцениваемые в размере эксабайт или больше. Такие компании, как Twitter, Facebook и YouTube, имеют миллиарды пользователей. Поэтому многие компании

services and products offered. Clients also adopt social media to get information about interesting services or goods (Najafabadi et al., 2015; Salehan & Kim, 2016; Xiang, Schwartz, Gerdes Jr, & Uysal, 2015).	используют социальные сети, чтобы поддерживать связь со своими клиентами и продвигать предлагаемые услуги и продукты. Клиенты также используют социальные сети для получения информации об интересных услугах или товарах (Najafabadi et al., 2015; Salehan & Kim, 2016; Xiang, Schwartz, Gerdes Jr, & Uysal, 2015).
The tremendous growth of social media with the users' generated data provide an excellent opportunity to mine valuable insights and understand better users' behaviors. This has motivated the development of big data solutions to solve many real-life issues (Najafabadi et al., 2015; Salehan & Kim, 2016; Xiang et al., 2015).	Огромный рост социальных сетей с данными, созданными пользователями, предоставляет отличную возможность добыть ценные идеи и лучше понять поведение пользователей. Это побудило к разработке решений для больших данных для решения многих реальных проблем (Najafabadi et al., 2015; Salehan & Kim, 2016; Xiang et al., 2015).
Generally, the term Big Data stands for data sets whose volume exceeds the capabilities of conventional tools to capture, manage, analyze and store data effectively (Bello-Orgaz, Jung, & Camacho, 2016; Garcna-Gil, Luengo, Garcna, & Herrera, 2019). The concept of Big data is characterized by the 5Vs (i.e., volume, velocity, variety, veracity, value).	Как правило, термин «большие данные» обозначает наборы данных, объем которых превышает возможности традиционных инструментов для эффективного сбора, управления, анализа и хранения данных (Bello-Orgaz, Jung & Camacho, 2016; Garcna-Gil, Luengo, Garcna & Herrera). 2019). Концепция больших данных характеризуется 5 В (т. Е. Объем, скорость, разнообразие, достоверность, ценность).
<ul style="list-style-type: none"> • Volume: This characteristic refers to massive volumes of data generated every second. Finding valuable insights through the exploration and analysis processes create serious problems for traditional tools. For instance, Flickr generates nearly 3.6 TB of data and Google handles approximately 20,000 TB each day (Zhang, Yang, Chen, & Li, 2018). Furthermore, as stated by the authors (McAfee, Brynjolfsson, Davenport, Patil, & Barton, 2012), in 2012, approximately 2.5 exabytes of data were generated each day. This amount of data doubles every 40 months. According to (Garcna-Gil et al., 2019; Garcna-Gil, Ramnrez-Gallego, Garcna, & Herrera, 2017), it is estimated that by 2020, the digital world will reach 44 zettabytes (i.e., 44 trillion gigabytes), which is 10 times larger than 4.4 zettabytes in 2013. 	<ul style="list-style-type: none"> • Объем: эта характеристика относится к огромным объемам данных, генерируемых каждую секунду. Поиск ценных идей в процессе исследования и анализа создает серьезные проблемы для традиционных инструментов. Например, Flickr генерирует почти 3,6 ТБ данных, а Google обрабатывает приблизительно 20 000 ТБ каждый день (Чжан, Ян, Чен и Ли, 2018). Кроме того, как утверждают авторы (McAfee, Brynjolfsson, Davenport, Patil, & Barton, 2012), в 2012 году примерно 2,5 эксабайта данных генерировалось каждый день. Это количество данных удваивается каждые 40 месяцев. Согласно (Garcna-Gil et al., 2019; Garcna-Gil, Ramnrez-Gallego, Garcna & Herrera, 2017), предполагается, что к 2020 году цифровой мир достигнет 44 зетабайтов (то есть 44 триллионов гигабайт), что в 10 раз больше, чем 4,4 зетабайта в 2013 году.
Velocity: The speed at which data are generated and should be processed. In particular, with the proliferation of digital devices such as sensors and smartphones, the data generated has witnessed an unprecedented rate of data	Скорость: скорость, с которой данные генерируются и должны быть обработаны. В частности, с распространением цифровых устройств, таких как датчики и смартфоны, сгенерированные

<p>growth, which poses serious challenges to handle streaming data and perform real-time analytics (Bello-Orgaz et al., 2016; Gandomi & Haider, 2015).</p>	<p>данные стали свидетелями беспрецедентной скорости роста данных, что создает серьезные проблемы для обработки потоковых данных и выполнения аналитики в реальном времени (Bello-Orgaz et al., 2016 ; Gandomi & Haider, 2015).</p>
<p>Variety: It indicates the various types of data that may be available in a structured, semi-structured, or unstructured format. Structured data constitutes the smallest percentage of all the existing data, where relational databases represent a typical example of structured data. Semi-structured refers to data that does not conform to strict standards like XML (Extensible Markup Language). The third type is unstructured data, which represents more than 75% of big data. It includes audio, text, video and images (Bello-Orgaz et al., 2016; Chen, Mao, & Liu, 2014; Gandomi & Haider, 2015; Zhang et al., 2018).</p>	<p>Разнообразие: указывает на различные типы данных, которые могут быть доступны в структурированном, полуструктурированном или неструктурированном формате. Структурированные данные составляют наименьший процент от всех существующих данных, где реляционные базы данных представляют собой типичный пример структурированных данных. Полуструктурированный относится к данным, которые не соответствуют строгим стандартам, таким как XML (расширяемый язык разметки). Третий тип - это неструктурированные данные, которые представляют более 75% больших данных. Он включает в себя аудио, текст, видео и изображения (Bello-Orgaz et al., 2016; Chen, Mao & Liu, 2014; Gandomi & Haider, 2015; Zhang et al., 2018).</p>
<p>Veracity: This characteristic was coined by IBM as the fourth V. It stands for the correctness and accuracy of data. In particular, with many forms of big data, quality and correctness are less controllable. For instance, the sentiments of users in social media are inherently uncertain because they involve human judgment. Nevertheless, they contain valuable information. Thus, the necessity to deal with uncertain and imprecise data is another aspect of Big Data, which should be addressed using the appropriate tools (Bello-Orgaz et al., 2016; Gandomi & Haider, 2015; Garcna-Gil et al., 2019).</p>	<p>Правдивость: эта характеристика была придумана IBM как четвертая V. Она обозначает правильность и точность данных. В частности, со многими формами больших данных качество и правильность менее контролируемы. Например, настроения пользователей в социальных сетях по своей природе неопределенны, поскольку они связаны с человеческим суждением. Тем не менее, они содержат ценную информацию. Таким образом, необходимость иметь дело с неопределенными и неточными данными является еще одним аспектом больших данных, которые должны решаться с использованием соответствующих инструментов (Bello-Orgaz et al., 2016; Gandomi & Haider, 2015; Garcna-Gil et al., 2019).</p>
<p>Value: Oracle added the so-called value as the fifth characteristic of Big data. At a simplistic level, big data has no intrinsic value. It becomes valuable only when we are able to derive the insights required to meet a particular need or address a problem. In other words, value is a key feature for any big data application as it allows generating useful business information (Bello-Orgaz et al., 2016; Lee, 2017).</p>	<p>Ценность: Oracle добавил так называемое значение в качестве пятой характеристики больших данных. На упрощенном уровне большие данные не имеют внутренней ценности. Это становится ценным только тогда, когда мы можем получить понимание, необходимое для удовлетворения конкретной потребности или решения проблемы. Другими словами, ценность является ключевой характеристикой любого приложения для работы с</p>

	<p>большими данными, поскольку оно позволяет генерировать полезную бизнес-информацию (Bello-Orgaz et al., 2016; Lee, 2017).</p>
<p>In the era of Big data, several companies around the world are using various solutions and techniques to analyze their huge amount of data in order to get meaningful insights. These techniques are called Big Data Analytics. In particular, the term big data analytics encompasses several algorithms, advanced statistics and applied analytics, which are used for various purposes such as prediction, classification, decision-making, and so on (Jimenez-Marquez, Gonzalez-Carrasco, Lopez-Cuadrado, & Ruiz-Mezcua, 2019; Najafabadi et al., 2015; Saggi & Jain, 2018; Wang, Kung, & Byrd, 2018a).</p>	<p>В эпоху Больших данных несколько компаний по всему миру используют различные решения и методы для анализа своих огромных объемов данных, чтобы получить осмысленное представление. Эти методы называются Big Data Analytics. В частности, термин «анализ больших данных» охватывает несколько алгоритмов, расширенную статистику и прикладную аналитику, которые используются для различных целей, таких как прогнозирование, классификация, принятие решений и т. Д. (Хименес-Маркес, Гонсалес-Карраско, Лопес-Куадрадо, & Ruiz-Mezcua, 2019; Najafabadi et al., 2015; Saggi & Jain, 2018; Wang, Kung & Byrd, 2018a).</p>
<p>More particularly, in addition to analyzing a large amount of data, Big Data Analytics creates serious challenges for machine learning techniques and data analysis task, including noisy data, highly distributed input data sources, high-dimensionality, limited labeled data, and so on. Furthermore, there are other real problems in Big Data Analytics such as data indexing, data storage, and information retrieval. As a result, more sophisticated data analysis and data management tools are necessary to handle the massive amount of data and deal with various real-world problems in the context of Big Data (Jimenez-Marquez et al., 2019; Najafabadi et al., 2015).</p>	<p>В частности, помимо анализа большого объема данных, Big Data Analytics создает серьезные проблемы для методов машинного обучения и задачи анализа данных, включая зашумленные данные, высокораспределенные источники входных данных, высокую размерность, ограниченные помеченные данные и т. Д. Кроме того, в Big Data Analytics есть и другие реальные проблемы, такие как индексация данных, хранение данных и поиск информации. В результате необходимы более сложные инструменты анализа данных и управления данными для обработки огромного объема данных и решения различных реальных проблем в контексте больших данных (Jimenez-Marquez et al., 2019; Najafabadi et al., 2015).</p>
<p>Due to the rapid development of social media, the huge amount of reviews generated caught the attention of several organizations, governments, businesses, and politicians to know the public opinion on various issues and understand the users' behaviors for specific purposes. Moreover, many users follow the opinions and feedback of others to decide the quality of a product or service before purchase. Therefore, sentiment analysis plays an essential role in decision-making tasks. It represents a practical technique, which is commonly used to deduce the sentiment polarity from people's opinions (Del Vecchio, Mele, Ndou, & Secundo, 2018; Jianqiang & Xiaolin, 2017; Ngai, Tao, & Moon, 2015; Pham & Le, 2018; Ragini, Anand, & Bhaskar, 2018; Rezaeinia, Rahmani, Ghodsi, &</p>	<p>В связи с быстрым развитием социальных сетей огромное количество обзоров привлекло внимание нескольких организаций, правительств, предприятий и политиков, чтобы узнать общественное мнение по различным вопросам и понять поведение пользователей в конкретных целях. Более того, многие пользователи следуют мнениям и отзывам других людей, чтобы определить качество продукта или услуги перед покупкой. Таким образом, анализ настроений играет важную роль в задачах принятия решений. Он представляет собой практический метод, который обычно используется для определения полярности</p>

<p>Veisi, 2019; Stieglitz, Mirbabaie, Ross, & Neuberger, 2018; Valdivia, Luzyn, & Herrera, 2017).</p>	<p>настроения на основе мнений людей (Del Vecchio, Mele, Ndou & Secundo, 2018; Jianqiang & Xiaolin, 2017; Ngai, Tao, & Moon, 2015; Pham & Le, 2018; Ragini, Anand & Bhaskar, 2018; Rezaeinia, Rahmani, Ghodsi & Veisi, 2019; Stieglitz, Mirbabaie, Ross & Neuberger, 2018; Valdivia, Luzyn & Herrera, 2017).</p>
<p>Deep learning (DL) stands for an extremely active research area in the machine learning (ML) community. It encompasses a set of learning algorithms, which are intended to automatically learn the hierarchical representations and extract the high-level features based on deep architectures. In particular, deep learning models have achieved remarkable success in various natural language processing tasks, including text classification and sentiment analysis (Chen et al., 2014; Rezaeinia et al., 2019; Zhang, Yang, & Chen, 2016).</p>	<p>Глубокое обучение (DL) означает чрезвычайно активную область исследований в сообществе машинного обучения (ML). Он включает в себя набор алгоритмов обучения, которые предназначены для автоматического изучения иерархических представлений и извлечения функций высокого уровня, основанных на глубоких архитектурах. В частности, модели глубокого обучения достигли значительных успехов в различных задачах обработки естественного языка, включая классификацию текста и анализ настроений (Chen et al., 2014; Rezaeinia et al., 2019; Zhang, Yang, & Chen, 2016).</p>
<p>Due to the different characteristics of big data, the design of an efficient big data system based on deep learning requires the consideration of many issues. In particular, because of the volume and the variety of big data sources, it is tricky to integrate effectively data collected from various distributed data sources. For example, more than 175 million tweets (i.e., unstructured data, which include images, videos, text, and so on) are posted by millions of user accounts in the whole world. In addition, it is necessary to store and handle the collected heterogeneous data efficiently. For instance, Facebook needs to manage, store and analyze more than 30 petabytes of data. Moreover, in order to take advantage of big data analytics, there is a need to analyze big data based on realtime, near real-time or batch processing. As a result, enhancing the performance of techniques used for a variety of tasks like classification, prediction, and visualization, is crucial for improving decision-making processes. Thus, many companies can benefit from the advantages of artificial intelligence with big data, increasing revenue by strengthening customer relationships (Hu, Wen, Chua, & Li, 2014; Jimenez-Marquez et al., 2019).</p>	<p>В связи с различными характеристиками больших данных разработка эффективной системы больших данных, основанной на глубоком обучении, требует рассмотрения многих вопросов. В частности, из-за большого объема и разнообразия источников больших данных сложно эффективно интегрировать данные, собранные из различных распределенных источников данных. Например, более 175 миллионов твитов (то есть неструктурированных данных, которые включают изображения, видео, текст и т. Д.) Размещены миллионами учетных записей пользователей во всем мире. Кроме того, необходимо эффективно хранить и обрабатывать собранные разнородные данные. Например, Facebook необходимо управлять, хранить и анализировать более 30 петабайт данных. Кроме того, чтобы воспользоваться преимуществами аналитики больших данных, необходимо анализировать большие данные на основе обработки в реальном времени, почти в реальном времени или пакетной обработки. В результате повышение эффективности методов, используемых для решения различных задач, таких как классификация, прогнозирование и визуализация, имеет решающее значение для улучшения процессов принятия решений. Таким образом, многие компании могут воспользоваться</p>

	преимуществами искусственного интеллекта при работе с большими данными, увеличивая доходы за счет укрепления отношений с клиентами (Ху, Вэнь, Чуа и Ли, 2014; Хименес-Маркес и др., 2019).
In this work, instead of directly applying DL models to real-world problems, one promising achievement is to improve the performance of the most powerful deep learning models for social big data analytics. In general, the concept of social big data analytics can be deemed as the intersection between big data analytics and social media. Its main objective is to take advantage of the efforts of the two fields for analyzing and extracting relevant knowledge from the huge amount of social media data. However, realtime social big data analytics aims to manage the complexity of conducting social big data analytics in real-time, to process unbounded data streams, which is the goal of our proposed framework.	В этой работе, вместо непосредственного применения моделей DL к реальным задачам, одним из многообещающих достижений является повышение производительности самых мощных моделей глубокого наклона для анализа больших данных в социальных сетях. В общем, понятие социальной аналитики больших данных можно рассматривать как пересечение аналитики больших данных и социальных сетей. Его главная цель - воспользоваться усилиями двух областей для анализа и извлечения соответствующих знаний из огромного количества данных социальных сетей. Однако анализ больших данных в реальном времени в социальных сетях направлен на управление сложностью проведения анализа больших данных в социальных сетях в реальном времени, на обработку неограниченных потоков данных, что является целью предлагаемой нами структуры.
The contributions of this paper are summarized as follows:	Вклад этой статьи резюмируется следующим образом:
Instead of using deep learning models to classify the reviews directly, we proposed a procedure based on fastText word embedding and Recurrent neural network variants for sentiment analysis, which is devoted to representing textual data efficiently.	Вместо того, чтобы использовать модели глубокого обучения для прямой классификации обзоров, мы предложили процедуру, основанную на встраивании слов fastText и рекуррентных вариантах нейронной сети для анализа настроений, которая посвящена эффективному представлению текстовых данных.
We designed an efficient strategy based on machine learning, which is tailored to improve the performance in terms of classification accuracy of three distributed Recurrent neural network variants, namely, Distributed Long Short-Term Memory (LSTM), Distributed Bidirectional Long Short-Term Memory (BiLSTM) and Distributed Gated Recurrent Unit (GRU) models.	Мы разработали эффективную стратегию, основанную на машинном обучении, которая предназначена для повышения производительности с точки зрения точности классификации трех вариантов распределенной рекуррентной нейронной сети, а именно распределенной долгосрочной краткосрочной памяти (LSTM), распределенной двунаправленной краткосрочной памяти (BiLSTM) и модели управляемых рекуррентных блоков (GRU).
We proposed a distributed intelligent system for real-time social big data analytics. It is based on a set of steps, which is dedicated not only to ingest, store, process, index, and visualize a large amount of information in real-time, but also it adopts a set of distributed machine learning and deep learning techniques for effective classification, prediction, and real-time analysis of customer behavior	Мы предложили распределенную интеллектуальную систему для анализа социальных данных в реальном времени. Он основан на наборе этапов, который предназначен не только для приема, хранения, обработки, индексации и визуализации большого объема информации в режиме реального времени, но также для

and public opinion.	принятия набора методов распределенного машинного обучения и глубокого обучения для эффективная классификация, прогнозирование и анализ поведения и общественного мнения потребителей в режиме реального времени.
We conducted a set of experimentations using two real-world data sets. The experimental results demonstrate that our proposal yields better classification accuracy than existing state-of-the-art methods. Moreover, it is able to improve the performance of several existing works.	Мы провели ряд экспериментов с использованием двух реальных наборов данных. Результаты эксперимента показывают, что наше предложение дает лучшую точность классификации, чем существующие современные методы. Более того, он способен улучшить выполнение нескольких существующих работ.
The rest of the paper is structured as follows. Next section outlines the related works. Section 3 describes Recurrent neural network (RNN) variants and fastText for sentiment analysis. Section 4 details our proposal. Section 5 describes the experiments. Finally, Section 6 concludes this paper.	Остальная часть статьи структурирована следующим образом. Следующий раздел описывает связанные работы. Раздел 3 описывает варианты рекуррентной нейронной сети (RNN) и fastText для анализа настроений. Раздел 4 детализирует наше предложение. Раздел 5 описывает эксперименты. Наконец, Раздел 6 завершает эту статью.
2. Related work	2. Связанная работа
With the advent of social networks, big data analytics played an important role in decision-making processes. Due to the remarkable success of Deep Learning (DL) approaches, various solutions have been proposed to cope with the challenges related to Natural language processing tasks.	С появлением социальных сетей аналитика больших данных сыграла важную роль в процессах принятия решений. Благодаря замечательному успеху подходов Deep Learning (DL) были предложены различные решения для решения задач, связанных с задачами обработки естественного языка.
In particular, several deep learning models for natural language processing have been designed based on employing word vector representations. For instance, Mikolov, Sutskever, Chen, Corrado, and Dean (2013) proposed the Word2Vec algorithm, a deep learning model for vector representations of the words, which is intended to convert words into meaningful vectors. Pennington, Socher, and Manning (2014) put forward a model for learning vector representations of words, namely, Global Vectors for Word Representation (GloVe). It is based on the global matrix factorization and local context window methods. Yu, Wang, Lai, and Zhang (2017) designed a word vector refinement model, which is intended to refine existing pre-trained word vectors such as Word2vec and GloVe. The key idea of the model is adjusting the word representations to capture both semantic and sentiment word vectors. Rezaeinia et al. (2019) developed an approach called Improved Word Vector (IWV), which is tailored to improve the accuracy of two pre-trained word embeddings, Word2Vec and GloVe vectors for sentiment analysis task. In particular, it employs four different approaches, POS	В частности, несколько моделей глубокого обучения для обработки естественного языка были разработаны на основе использования представлений векторов слов. Например, Mikolov, Sutskever, Chen, Corrado и Dean (2013) предложили алгоритм Word2Vec, модель глубокого обучения для векторных представлений слов, которая предназначена для преобразования слов в значимые векторы. Pennington, Socher и Manning (2014) выдвинули модель для изучения векторных представлений слов, а именно, Глобальные векторы для представления слов (GloVe). Он основан на факторизации глобальной матрицы и методах локального контекстного окна. Yu, Wang, Lai и Zhang (2017) разработали модель уточнения вектора слов, которая предназначена для уточнения существующих предварительно обученных векторов слов, таких как Word2vec и GloVe. Ключевой идеей модели является корректировка представлений слов для

<p>tagging, Word2Vec with GloVe, word position and lexicon-based approach. Bojanowski, Grave, Joulin, and Mikolov (2017) designed an approach for learning word representations by taking into account subword information. The key idea is to incorporate character n-grams into the skip-gram model, where each word is considered as a bag of character n-grams. Each character is represented as a vector representation.</p>	<p>захвата как семантических, так и чувствительных векторов слов. Резейния и соавт. (2019) разработали подход под названием «Улучшенный вектор слов» (IWV), который предназначен для повышения точности двух предварительно обученных вложений слов, векторов Word2Vec и GloVe для задачи анализа настроений. В частности, он использует четыре различных подхода: POS-теги, Word2Vec с GloVe, положение слов и подход, основанный на лексиконах. Bojanowski, Grave, Joulin и Mikolov (2017) разработали подход для изучения представлений слов с учетом информации подслов. Основная идея заключается в том, чтобы включить n-грамм символов в модель скип-граммы, где каждое слово рассматривается как пакет n-грамм символов. Каждый символ представлен в виде векторного представления.</p>
<p>However, other researchers have tended to focus on the problem of sentiment analysis and text classification based on word embedding with deep learning. Kim (2014) proposed three deep learning models, which are designed based on Convolution Neural Network (CNN) and pre-trained word2vec vectors for sentence classification. Kalchbrenner, Grefenstette, and Blunsom (2014) proposed a deep learning architecture for the semantic modeling of sentences, called dynamic convolutional neural network (DCNN). It is devoted to analyzing and representing the semantic content of sentences for classification tasks. Chen, Xu, He, and Wang (2017) designed an approach for sentence-level sentiment classification. It is based on two steps. First, it classifies sentences into different types. Second, each group of sentences is fed into a convolutional neural network model for performing sentiment analysis task. Jianqiang, Xiaolin, and Xuejun (2018) introduced an approach, which aims to combine glove word embeddings with n-grams features and the polarity score features to represent the features of tweets. Then, a deep CNN model is adopted to process these features and perform the sentiment classification task. While, Alharbi and de Doncker (2019) developed a convolutional neural network model, which is designed to incorporate user behavioral information according to each tweet.</p>	<p>Тем не менее, другие исследователи, как правило, сосредоточены на проблеме анализа настроений и классификации текста на основе вложения слов с глубоким изучением. Ким (2014) предложил три модели глубокого обучения, которые разработаны на основе Convolution Neural Network (CNN) и предварительно обученных векторов word2vec для классификации предложений. Kalchbrenner, Grefenstette и Blunsom (2014) предложили архитектуру глубокого обучения для семантического моделирования предложений, названную динамической сверточной нейронной сетью (DCNN). Он посвящен анализу и представлению семантического содержания предложений для задач классификации. Chen, Xu, He и Wang (2017) разработали подход для классификации настроений на уровне предложений. Он основан на двух шагах. Во-первых, он классифицирует предложения на разные типы. Во-вторых, каждая группа предложений подается в сверточную модель нейронной сети для выполнения задачи анализа настроений. Jianqiang, Xiaolin и Xuejun (2018) представили подход, который направлен на объединение вложений слов в перчатках с функциями n-грамм и функциями оценки полярности для представления особенностей твитов. Затем применяется глубокая модель CNN для обработки этих признаков и выполнения задачи классификации настроений. В то время как Alharbi и de Doncker (2019) разработали</p>

	сверточную модель нейронной сети, которая предназначена для включения информации о поведении пользователей в соответствии с каждым твитом.
On the other hand, Iyyer, Manjunatha, Boyd-Graber, and Daumй III (2015) proposed a deep neural network for text classification, namely, deep averaging network (DAN). It is tailored to process an unweighted average of word vectors through multiple hidden layers before performing the classification task. Socher, Pennington, Huang, Ng, and Manning (2011) introduced a framework based on semi-supervised recursive autoencoders for predict sentence-level sentiment distributions. Socher et al. (2013) developed an approach called Recursive Neural Tensor Network (RNTN). The intuition behind this model is that it calculates the compositional vector representations for phrases. Then, the representations will be utilized as features for classifying each phrase.	С другой стороны, Iyyer, Manjunatha, Boyd-Graber и Daumй III (2015) предложили глубокую нейронную сеть для классификации текста, а именно сеть глубокого усреднения (DAN). Он предназначен для обработки невзвешенного среднего вектора слов через несколько скрытых слоев перед выполнением задачи классификации. Socher, Pennington, Huang, Ng и Manning (2011) представили структуру, основанную на полуконтролируемых рекурсивных автоэнкодерах для прогнозирования распределения настроений на уровне предложений. Socher et al. (2013) разработали подход под названием «Рекурсивная нейронная тензорная сеть» (RNTN). Интуиция за этой моделью состоит в том, что она вычисляет композиционные векторные представления для фраз. Затем представления будут использоваться как функции для классификации каждой фразы.
The advent of sequence modeling has inspired many researchers to design effective approaches for sentiment analysis. For example, Wang, Liu, Chengjie, Wang, and Wang (2015) proposed Long Short-Term Memory (LSTM) recurrent network for twitter sentiment prediction. It aims to compose word representations through a flexible compositional function. Tai, Socher, and Manning (2015) introduced an approach called Tree-LSTM. It represents a generalization of the conventional LSTM architecture to tree-structured network topology. Wang, Yu, Lai, and Zhang (2016) developed a method called regional CNN-LSTM model for dimensional sentiment analysis. It is designed based on regional CNN and LSTM. In particular, unlike a conventional CNN which process the whole text as input, the regional CNN considers an input text as a set of regions (i.e., sentences), therefore, it treats each sentence separately. Then, LSTM model is adopted to consider long-distance dependency across sentences in the prediction process. Vosoughi, Vijayaraghavan, and Roy (2016) proposed an approach for learning tweet embeddings. called Tweet2Vec. It is designed based on character-level CNN-LSTM encoder-decoder. Tien and Le (2017) introduced an approach based on CNN and LSTM, which aims to synthesize feature vectors for performing sentiment analysis. Liu and Guo (2019) developed an approach, called attentionbased bidirectional long short-term memory with convolution	Появление моделирования последовательностей вдохновило многих исследователей на разработку эффективных подходов для анализа настроений. Например, Wang, Liu, Chengjie, Wang и Wang (2015) предложили рекуррентную сеть с кратковременной памятью (LSTM) для прогнозирования настроений в Твиттере. Это стремится составить представления слова через гибкую композиционную функцию. Tai, Socher и Manning (2015) представили подход под названием Tree-LSTM. Он представляет собой обобщение традиционной архитектуры LSTM на топологию сети с древовидной структурой. Wang, Yu, Lai и Zhang (2016) разработали метод, названный региональной моделью CNN-LSTM для анализа многомерных настроений. Он разработан на основе региональных CNN и LSTM. В частности, в отличие от обычной CNN, которая обрабатывает весь текст в качестве входных данных, региональная CNN рассматривает входной текст как набор областей (то есть предложений), поэтому обрабатывает каждое предложение отдельно. Затем принимается модель LSTM для учета зависимости на расстоянии между предложениями в процессе прогнозирования. Vosoughi, Vijayaraghavan и Roy (2016)

<p>layer (AC-BiLSTM) for sentiment analysis. This approach is based on a bidirectional LSTM, attention mechanism and the convolutional layer. Specifically, the convolutional layer is used to extract higherlevel phrase representations. BiLSTM is adopted to access the preceding and succeeding context representations. While the attention mechanism is intended to give focus on the information produced by the hidden layers of BiLSTM.</p>	<p>предложили подход для изучения встраивания твитов. называется Tweet2Vec. Он разработан на основе кодировщика-декодера CNN-LSTM уровня символов. Tien and Le (2017) представили подход, основанный на CNN и LSTM, который направлен на синтез векторов признаков для выполнения анализа настроений. Liu и Guo (2019) разработали подход, называемый двунаправленной долговременной памятью на основе внимания со слоем свертки (AC-BiLSTM) для анализа настроений. Этот подход основан на двунаправленном LSTM, механизме внимания и сверточном слое. В частности, сверточный слой используется для извлечения представлений фраз более высокого уровня. BiLSTM используется для доступа к предыдущему и последующему контекстным представлениям. При этом механизм внимания предназначен для того, чтобы сосредоточить внимание на информации, создаваемой скрытыми слоями BiLSTM.</p>
<p>3. Recurrent neural network variants with fastText for sentiment analysis</p>	<p>3. Периодические варианты нейронной сети с FastText для анализа настроений</p>
<p>This section presents the word embedding technique called fastText, with Recurrent neural network, Long short-term memory, Bidirectional long short-term memory and Gated recurrent unit methods for Sentiment analysis.</p>	<p>В этом разделе представлена методика встраивания слов fastText с рекуррентной нейронной сетью, длинной кратковременной памятью, двунаправленной краткосрочной памятью и рекуррентными методами Gated для анализа настроений.</p>
<p><i>3.1. FastText: word representation learning</i></p>	<p>3.1. FastText: обучение представлению слов</p>
<p>With the success of deep learning, the models based on neural networks have become increasingly popular to convert words into meaningful vectors. These continuous representations of words stand for the fundamental building blocks for many Natural Language Processing (NLP) applications, such as clustering, information retrieval and text classification (Mikolov, Grave, Bojanowski, Puhersch, & Joulin, 2018; Rezaeinia et al., 2019).</p>	<p>С успехом глубокого обучения модели, основанные на нейронных сетях, становятся все более популярными для преобразования слов в значимые векторы. Эти непрерывные представления слов означают фундаментальные строительные блоки для многих приложений обработки естественного языка (NLP), таких как кластеризация, поиск информации и классификация текста (Миколов, Грейв, Бояновский, Пурш и Йоулин, 2018; Rezaeinia et al., 2019).</p>
<p>The fastText is one of the most popular word embedding techniques based on neural networks, which is created by Bojanowski et al. at Facebook's AI Research lab (Mikolov et al., 2018). It is tailored to learn high-quality representations of words while considering the morphology.</p>	<p>FastText - это одна из самых популярных техник встраивания слов на основе нейронных сетей, созданная Bojanowski et al. в исследовательской лаборатории ФИ Facebook (Миколов и др., 2018). Он предназначен для изучения высококачественных представлений слов с учетом морфологии.</p>
<p>In general, fastText is based on the continuous skipgram model introduced by</p>	<p>В общем, fastText основан на модели непрерывных скипграмм,</p>

Mikolov et al. (2013) and subword model.	представленной Mikolov et al. (2013) и модель подслов.
3.1.1. Skipgram model with negative sampling	3.1.1. Модель скипграмм с отрицательной выборкой
Let W be the size of the word vocabulary, where each word is identified by its index denoted $w \in \{1, \dots, W\}$. Given a training corpus defined as a sequence of words w_1, \dots, w_T , the main goal of the skipgram algorithm is to maximize the following objective function:	Пусть W - размер словарного запаса слова, где каждое слово идентифицируется по индексу, обозначенному $w \in \{1, \dots, W\}$. Учитывая тренировочный корпус, определенный как последовательность слов w_1, \dots, w_T , главная цель алгоритма скипграммы - максимизировать следующую целевую функцию:
$O = \sum_{t=1}^T \sum_{c \in C_t} \log(p(w_c w_t)) \quad (1)$	
where C_t represents the context, which is defined as the set of indices of words surrounding the word w_t .	где C_t представляет контекст, который определяется как набор индексов слов, окружающих слово w_t .
The problem of predicting context words can be considered as a set of independent binary classification tasks. The objective is to predict the presence or absence of context words. In particular, for the word at position t , all context words are considered as positive examples, and negatives are sampled randomly from the vocabulary. As a result, the objective function can be defined as follows:	Проблема прогнозирования контекстных слов может рассматриваться как совокупность независимых задач двоичной классификации. Цель состоит в том, чтобы предсказать наличие или отсутствие контекстных слов. В частности, для слова в позиции t все контекстные слова рассматриваются как положительные примеры, а отрицательные значения выбираются случайным образом из словаря. В результате целевая функция может быть определена следующим образом:
$\log(1 + e^{-s(w_t, w_c)}) + \sum_{n \in N_{t,c}} \log(1 + e^{s(w_t, n)}) \quad (2)$	
where $N_{t,c}$ stands for a set of negative examples sampled from the vocabulary.	где N_t , c обозначает набор отрицательных примеров, взятых из словаря.
Let u_{wt} and v_{wc} , two vectors corresponding to the words w_t and w_c , respectively. The function $s(w_t, w_c)$ can be computed as the scalar product between word and context vectors as follows:	Пусть u_{wt} и v_{wc} , два вектора, соответствующие словам w_t и w_c соответственно. Функция $s(w_t, w_c)$ может быть вычислена как скалярное произведение между векторами слова и контекста следующим образом:
$s(w_t, w_c) = u_{w_t}^T v_{w_c} \quad (3)$	
3.1.2. Subword model	3.1.2. Модель подслов
In this step, each word is considered as a bag of character n-grams. Let D be a dictionary of n-grams of size $ D $. For each word w , Dw denotes the set of n-grams appearing in w , where each vector representation z_d is associated to each n-gram	На этом этапе каждое слово рассматривается как мешок символов n-грамм. Пусть D словарь из n-грамм размера $ D $. Для каждого слова w Dw обозначает набор n-грамм, фигурирующих в w , где

denoted d . Thus, each word w is represented by the sum of the vector representations of its n -grams.	каждое векторное представление z_d связано с каждым n -граммом, обозначенным как d . Таким образом, каждое слово w представлено суммой векторных представлений его n -грамм.
$s(w, c) = \sum_{d \in D_w} z_d^T v_c \quad (4)$	
This model facilitate sharing the representations across words, which allows to learn reliable representation for rare words (Bojanowski et al., 2017).	Эта модель облегчает разделение представлений между словами, что позволяет выучить надежное представление для редких слов (Bojanowski et al., 2017).
<i>3.2. Deep learning architectures for sequence modeling</i>	3.2. Архитектура глубокого обучения для моделирования последовательностей
<i>3.2.1. Recurrent neural network</i>	3.2.1. Рекуррентная нейронная сеть
Recurrent neural network (RNN) is a class of neural network architectures which is devoted for modeling sequential data. In particular, it is extremely useful for several natural language processing tasks such as sentiment analysis, text classification, speech-to-text, machine translation, and so on.	Рекуррентная нейронная сеть (RNN) - это класс архитектур нейронных сетей, предназначенный для моделирования последовательных данных. В частности, это чрезвычайно полезно для нескольких задач обработки естественного языка, таких как анализ настроений, классификация текста, речевой текст, машинный перевод и так далее.
Let $V = (v_1, \dots, v_N)$ be an input sequence, which represents a review composed of N words as a set of N fastText vector representations. Each word w_i is represented by a D dimensional vector $v_i \in \mathbb{R}^D$.	Пусть $V = (v_1, \dots, v_N)$ - входная последовательность, которая представляет обзор, состоящий из N слов, как набор из N векторных представлений fastText. Каждое слово w_i представлено D -мерным вектором $v_i \in \mathbb{R}^D$.
The RNN model iterates over the N words to calculate the hidden states represented as (h_1, \dots, h_N) , and produces the output. More precisely, at each time step t , the RNN model takes both the word embedding vector $v_t \in \mathbb{R}^D$ and the previous hidden state $h_{t-1} \in \mathbb{R}^M$ as input. It calculates the current hidden state $h_t \in \mathbb{R}^M$ based on the following equation:	Модель RNN выполняет итерацию по N словам для вычисления скрытых состояний, представленных как (h_1, \dots, h_N) , и выдает выходные данные. Точнее, на каждом временном шаге t модель RNN принимает как вектор внедрения слова $v_t \in \mathbb{R}^D$, так и предыдущее скрытое состояние $h_{t-1} \in \mathbb{R}^M$ в качестве входных данных. Он вычисляет текущее скрытое состояние $h_t \in \mathbb{R}^M$ на основе следующего уравнения:
$h_t = H(W_h v_t + U_h h_{t-1} + b_h) \quad (5)$	
where H denotes a non-linear activation function such as sigmoid, hyperbolic tangent function. The symbols $W_h \in \mathbb{R}^{M \times D}$ and $U_h \in \mathbb{R}^{M \times M}$ refer to the weight matrices, $b_h \in \mathbb{R}^M$ represents the bias term, M denotes the number of hidden units (Graves, Mohamed, & Hinton, 2013; Kim, Jernite, Sontag, & Rush, 2016; Sutskever, Martens, & Hinton, 2011; Tai et al., 2015).	где H обозначает нелинейную функцию активации, такую как сигмоидальная, гиперболическая касательная функция. Символы $W_h \in \mathbb{R}^{M \times D}$ и $U_h \in \mathbb{R}^{M \times M}$ относятся к весовым матрицам, $b_h \in \mathbb{R}^M$ представляет термин смещения, M обозначает количество скрытых единиц (Graves, Mohamed & Hinton, 2013; Kim, Jernite,

	Sontag, & Rush, 2016 ; Sutskever, Martens, & Hinton, 2011; Tai et al., 2015).
Next, based on the hidden state h_t , the predicted output y_t is defined as follows:	Затем на основе скрытого состояния h_t прогнозируемый выходной сигнал y_t определяется следующим образом:
$y_t = \text{softmax}(W_y h_t + b_y) \quad (6)$	
where $W_y \times C M$ is the weight matrix, C is the number of output classes, b_y refers to the bias term.	где $W_y \times C M$ - весовая матрица, C - количество выходных классов, относится к термину смещения.
In general, for training the RNN model, an extension of the backpropagation algorithm, known as Backpropagation Through Time (BPTT), is adopted to perform this task. However, in practical situations, the recurrent neural network model faces the vanishing/ exploding gradients, a real problem that negatively affects the RNN performance, which prevents it from learning long-term dependencies (Kim et al., 2016; Mohammadi, Al-Fuqaha, Sorour, & Guizani, 2018).	В общем, для обучения модели RNN для выполнения этой задачи используется расширение алгоритма обратного распространения, известное как обратное распространение через время (BPTT). Однако в практических ситуациях рекуррентная модель нейронной сети сталкивается с исчезающими / взрывающимися градиентами, реальной проблемой, которая негативно влияет на производительность RNN, что не позволяет ей изучать долгосрочные зависимости (Kim et al., 2016; Mohammadi, Al-Fuqaha Sorour & Guizani, 2018).
3.2.2. <i>Long short-term memory</i>	3.2.2. Долгосрочная кратковременная память
Long Short-Term Memory network (LSTM) stands for a special type of recurrent neural network architecture, which has shown great promise in several natural language processing tasks. It is mainly tailored to learn long-term dependencies by adopting memory cells to store information. In addition, a system of gating units is employed to control the information flow into and out of the cell. As a result, LSTM architecture is able to overcome the vanishing gradient, and work better than the standard recurrent neural network (Graves & Schmidhuber, 2005; Greff, Srivastava, Koutnнк, Steunebrink, & Schmidhuber, 2017; Hochreiter & Schmidhuber, 1997; Kim et al., 2016; Mohammadi et al., 2018; Song, Park, & Shin, 2019).	Сеть с долгосрочной кратковременной памятью (LSTM) представляет собой особый тип рекуррентной архитектуры нейронной сети, которая показала большие перспективы в нескольких задачах обработки естественного языка. Она в основном предназначена для изучения долгосрочных зависимостей, используя ячейки памяти для хранения информации. Кроме того, система стробирующих устройств используется для управления потоком информации в ячейку и из нее. В результате архитектура LSTM способна преодолеть исчезающий градиент и работать лучше, чем стандартная рекуррентная нейронная сеть (Graves & Schmidhuber, 2005; Greff, Srivastava, Koutnнк, Steunebrink, & Schmidhuber, 2017; Hochreiter & Schmidhuber, 1997; Kim et al., 2016; Mohammadi et al., 2018; Song, Park & Shin, 2019).
Let N be the number of LSTM cells. One LSTM cell has a hidden state h_t , a memory cell c_t , and three different gates, namely, input gate i_t , forget gate f_t , and an output gate o_t .	Пусть N будет количеством ячеек LSTM. Одна ячейка LSTM имеет скрытое состояние h_t , ячейку памяти c_t и три разных шлюза, а именно: входной шлюз i_t , забытый шлюз f_t и выходной шлюз.

<p>Forget gate: it is devoted to controlling how much of the previous long-term state is read into the current cell state.</p> <ul style="list-style-type: none"> • Input Gate: it controls how much of the current input is added into the cell state. • Output Gate: it decides which parts of the long term state should be in the next hidden state. 	<p>Забывший шлюз: он предназначен для контроля того, сколько из предыдущего долгосрочного состояния считывается в текущее состояние ячейки.</p> <ul style="list-style-type: none"> • Входной шлюз: он контролирует, сколько текущего ввода добавляется в состояние ячейки. • Выходной шлюз: он решает, какие части долгосрочного состояния должны находиться в следующем скрытом состоянии.
Each one of the gates adopts the sigmoid as an activation function, and produces a value in the interval $[0, 1]$.	Каждый из шлюзов принимает сигмоид в качестве функции активации и выдает значение в интервале $[0, 1]$.
$\begin{aligned} i_t &= \sigma(W_i v_t + U_i h_{t-1} + b_i) \\ f_t &= \sigma(W_f v_t + U_f h_{t-1} + b_f) \\ o_t &= \sigma(W_o v_t + U_o h_{t-1} + b_o) \end{aligned} \quad (7)$	
where $\sigma(\cdot)$ represents the sigmoid function. $W_i, W_f, W_o, U_i, U_f, U_o$ are the weight matrices for the gates. b_i, b_f, b_o are the bias weights.	где $\sigma(\cdot)$ представляет сигмовидную функцию. $W_i, W_f, W_o, U_i, U_f, U_o$ - весовые матрицы для ворот. b_i, b_f, b_o - веса смещения.
Generally, at the time step t , the LSTM architecture takes as input the word representation vector v_t , the previous hidden state h_{t-1} and the previous memory cell vector c_{t-1} . It calculates h_t, c_t as follows:	Как правило, на этапе t архитектура LSTM принимает в качестве входных данных вектор представления слов v_t , предыдущего скрытого состояния h_{t-1} и предыдущего вектора ячеек памяти c_{t-1} . Он вычисляет h_t, c_t следующим образом:
$\begin{aligned} g_t &= \tanh(W_g v_t + U_g h_{t-1} + b_g) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \quad (8)$	
where the notation \odot denotes the element-wise multiplication, $\tanh(\cdot)$ refers to the hyperbolic tangent function, W_g and U_g are the weight matrices, b_g is the bias term.	где обозначение \odot обозначает поэлементное умножение, $\tanh(\cdot)$ относится к гиперболической касательной функции, W_g и U_g - весовые матрицы, b_g - термин смещения.
3.2.3. Bidirectional long short-term memory	3.2.3. Двухнаправленная долговременная память
Bidirectional Long Short-Term Memory (BiLSTM) represents an extension of the unidirectional LSTM. Specifically, it employs two LSTMs. The first one is the forward LSTM denoted by $LSTM_f$. It processes the input sequence from left to right. The second is the backward LSTM, which is denoted by $LSTM_b$. It processes the sequence in reverse order, from right to left (Chen et al., 2017; Graves et al., 2013; Lin, Xu, Luo, & Zhu, 2017; Tai et al., 2015).	Двухнаправленная долговременная кратковременная память (BiLSTM) представляет собой расширение однонаправленной LSTM. В частности, он использует два LSTM. Первый - это прямой LSTM, обозначаемый $LSTM_f$. Обработывает последовательность ввода слева направо. Вторым является обратный LSTM, который обозначается $LSTM_b$. Он обрабатывает последовательность в обратном порядке справа налево (Chen et al., 2017; Graves et al., 2013; Lin, Xu, Luo, & Zhu, 2017; Tai et al., 2015).

	2015).
The key idea behind the Bidirectional LSTM model is to exploit both future and past context. At each time step t , it generates the hidden state by concatenating the forward hidden state ht and the backward hidden state ht defined as follows:	Основная идея двунаправленной модели LSTM заключается в использовании как будущего, так и прошлого контекста. На каждом временном шаге t он создает скрытое состояние путем объединения скрытого прямого состояния ht и скрытого обратного состояния ht , определенного следующим образом:
$\begin{aligned}\vec{h}_t &= \vec{LSTM}(v_t, \vec{h}_{t-1}) \\ \overleftarrow{h}_t &= \overleftarrow{LSTM}(v_t, \overleftarrow{h}_{t+1})\end{aligned}\quad (9)$	
where the forward hidden state ht is calculated based on the previous hidden state $ht-1$ and the input vt , while the backward hidden state ht is computed based on the future hidden state $ht+1$ and vt .	где скрытое прямое состояние ht вычисляется на основе предыдущего скрытого состояния $ht-1$ и входных данных vt , тогда как скрытое обратное состояние ht вычисляется на основе будущего скрытого состояния $ht+1$ и vt .
3.2.4. Gated recurrent unit	3.2.4. Управляемые рекуррентные блоки
Gated recurrent unit network stands for a variant of the standard LSTM network. It designed to simplify the complex architecture of LSTM networks by reducing the number of trainable parameters in each cell. It has proven its ability to provide better performance than other existing models.	Управляемый рекуррентный блок сети выступает за вариант стандартной сети LSTM. Он предназначен для упрощения сложной архитектуры сетей LSTM за счет уменьшения количества обучаемых параметров в каждой ячейке. Он доказал свою способность обеспечивать лучшую производительность, чем другие существующие модели.
As a traditional LSTM unit, Gated Recurrent Unit (GRU) adopts a gating mechanism to control information flow inside the unit. It has only two gates for controlling the hidden state, known as the reset gate rt and the update gate ut (Cho et al., 2014; Chung, Gulcehre, Cho, & Bengio, 2014).	Как традиционное устройство LSTM, управляемый рекуррентный блок (GRU) использует стробирующий механизм для управления информационным потоком внутри устройства. Он имеет только два шлюза для управления скрытым состоянием, известные как ворота сброса rt и ворота обновления ut (Cho et al., 2014; Chung, Gulcehre, Cho, & Bengio, 2014).
Mathematically, these two gates are defined as follows:	Математически эти два элемента определяются следующим образом:
$\begin{aligned}r_t &= \sigma(W_r v_t + U_r h_{t-1} + b_r) \\ u_t &= \sigma(W_u v_t + U_u h_{t-1} + b_u)\end{aligned}\quad (10)$	
At the time step t , the activation ht and the candidate activation h^t of GRU are computed as follows:	На шаге времени t активация ht и возможная активация h^t GRU вычисляются следующим образом:

$\hat{h}_t = \tanh(W_{\hat{h}}v_t + U_{\hat{h}}(r_t \odot h_{t-1}) + b_{\hat{h}})$ $h_t = u_t \hat{h}_t + (1 - u_t) \odot h_{t-1} \quad (11)$	
<p>where $W_r, W_u, W_{\hat{h}}, U_r, U_u, U_{\hat{h}}$ are the weight matrices. $b_r, b_h, b_{\hat{h}}$ are the bias terms</p>	<p>где $W_r, W_u, W_{\hat{h}}, U_r, U_u, U_{\hat{h}}$ - весовые матрицы. $b_r, b_h, b_{\hat{h}}$ являются терминами смещения</p>
<p>4. Our proposal</p>	<p>4. Наше предложение</p>
<p>In this section, we present a distributed intelligent system for social big data analytics, and our proposal for improving distributed RNN variants. The key idea is to perform real-time analysis, improve the classification performance of the most powerful RNN models, and handle large-scale data sets using parallel and distributed training.</p>	<p>В этом разделе мы представляем распределенную интеллектуальную систему для анализа больших данных в социальных сетях и наше предложение по улучшению распределенных вариантов RNN. Основная идея заключается в проведении анализа в реальном времени, повышении эффективности классификации наиболее мощных моделей RNN и обработке крупномасштабных наборов данных с использованием параллельного и распределенного обучения.</p>
<p><i>4.1. Proposed distributed intelligent architecture for real-time big data analytics</i></p>	<p>4.1. Предлагаемая распределенная интеллектуальная архитектура для анализа больших данных в реальном времени</p>
<p>To deal with Big Data issues, we have proposed a distributed intelligent system for real-time stream processing, which is based on a set of tools devoted to handling the large amount of data and perform big data analytics in an efficient and effective way.</p>	<p>Для решения проблем больших данных мы предложили распределенную интеллектуальную систему для обработки потоков в реальном времени, которая основана на наборе инструментов, предназначенных для обработки большого объема данных и эффективного и действенного анализа больших данных.</p>
<p>The system combines multiple layers into a single architecture. This pipeline architecture aims to run a sequence of steps needed for processing and analyzing data. As illustrated in Fig. 1, the system consists of a set of stages organized as follows:</p>	<p>Система объединяет несколько уровней в единую архитектуру. Эта конвейерная архитектура направлена на выполнение последовательности шагов, необходимых для обработки и анализа данных. Как показано на рис. 1, система состоит из набора этапов, организованных следующим образом:</p>
<ol style="list-style-type: none"> 1. Big Data acquisition 2. Big Data storage layer 3. Big Data ingestion layer 4. Big Data processing layer 5. Distributed deep learning for natural language processing layer 6. Distributed machine learning layer 7. Big Data indexing layer 8. Real-time visualization layer 	<ol style="list-style-type: none"> 1. Сбор больших данных 2. Уровень хранения больших данных 3. Уровень приема больших данных 4. Уровень обработки больших данных 5. Распределенное глубокое обучение для уровня обработки естественного языка 6. Уровень распределенного машинного обучения 7. Уровень индексации больших данных 8. В режиме реального времени слой визуализации

Fig. 1. Flowchart of the distributed real-time system for social Big Data

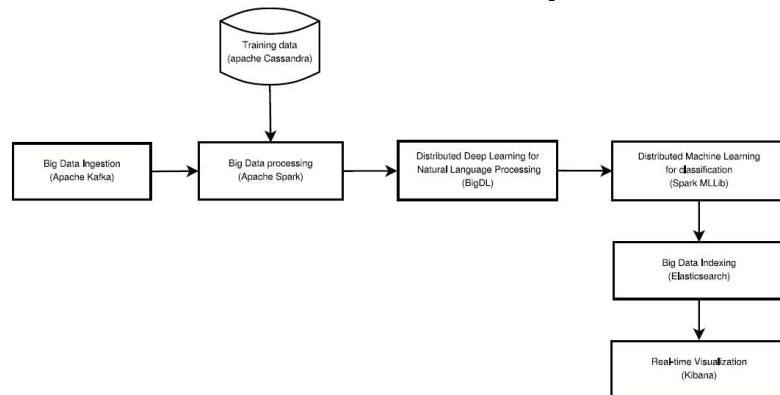


Рис. 1. Блок-схема распределенной системы реального времени для социальной аналитики больших данных.

Analytics.

4.1.1. Big data acquisition

This layer is devoted to collecting data from several internal and external sources. The data can be available as structured data, semi-structured data, and unstructured data. In this work, as a case study, the data to be processed is textual data from several social media sources, i.e., users' reviews representing the opinions of users on several topics, to provide the insights required for achieving specific objectives, such as solving serious business problems, real-time analysis of customer behavior and public opinion.

4.1.1. Сбор больших данных

Этот слой предназначен для сбора данных из нескольких внутренних и внешних источников. Данные могут быть доступны в виде структурированных данных, полуструктурированных данных и неструктурированных данных. В этой работе, например, в качестве тематического исследования, обрабатываемые данные представляют собой текстовые данные из нескольких источников в социальных сетях, т. Е. Отзывы пользователей, представляющие мнения пользователей по нескольким темам, для предоставления информации, необходимой для достижения конкретных целей, таких как решение серьезные бизнес-проблемы, анализ поведения клиентов и общественного мнения в режиме реального времени.

4.1.2. Big data ingestion layer

In the big data era, real-time information is generated continuously. To deliver the data to several types of receivers, an effective way is needed to perform this task. This layer is responsible to quickly deliver information collected from various data sources for facilitating the consumption of millions of messages in the pipeline architecture. Therefore, we have adopted Apache Kafka, a distributed streaming platform, which was developed at LinkedIn. It is a fault-tolerant, scalable, real-time publish-subscribe messaging system.

4.1.2. Слой приема больших данных

В эпоху больших данных информация в реальном времени генерируется непрерывно. Для доставки данных нескольким типам приемников необходим эффективный способ выполнения этой задачи. Этот уровень отвечает за быструю доставку информации, собранной из различных источников данных, для облегчения потребления миллионов сообщений в конвейерной архитектуре. Поэтому мы приняли Apache Kafka, платформу потокового вещания, которая была разработана в LinkedIn. Это отказоустойчивая, масштабируемая система обмена сообщениями с подпиской в режиме реального времени.

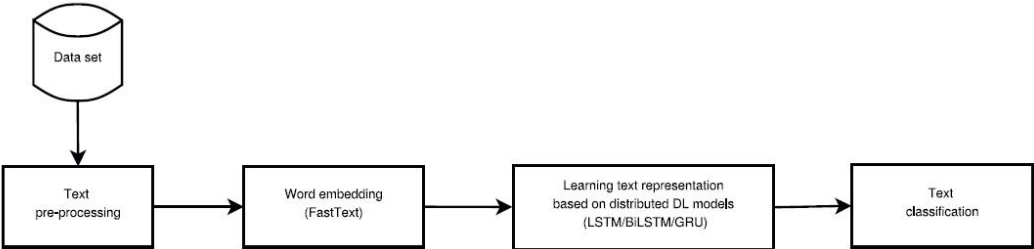
In general, Kafka is based on a set of concepts such as topic, producer, consumer,

В целом, Kafka основан на наборе понятий, таких как тема,

and broker. A topic represents a stream of records of a particular type. A producer enables an application to publish a stream of messages to one or many topics. All the published records are stored at a set of servers called brokers. Whereas a consumer enables an application to subscribe to at least one or many topics and handle the stream of messages produced (Jan et al., 2019; Kafka, 2019; Kreps, Narkhede, Rao et al., 2011).	производитель, потребитель и брокер. Тема представляет собой поток записей определенного типа. Производитель позволяет приложению публиковать поток сообщений на одну или несколько тем. Все опубликованные записи хранятся на наборе серверов, называемых брокерами. Принимая во внимание, что потребитель позволяет приложению подписаться как минимум на одну или несколько тем и обрабатывать поток создаваемых сообщений (Jan et al., 2019; Kafka, 2019; Kreps, Narkhede, Rao et al., 2011).
<i>4.1.3. Big data storage layer</i>	4.1.3. Слой хранения больших данных
This layer is intended to store the data needed for training and testing the distributed deep learning models. As NoSQL represents a category of databases tailored for handling Big Data. We have adopted Apache Cassandra, a distributed NoSQL database management system, originally developed by Facebook. It is devoted to managing large amounts of data spread out across many nodes, possibly different data centers. It offers several advantages including scalability, high availability, fault-tolerance without compromising performance (Cassandra, 2019; Lakshman & Malik, 2010).	Этот слой предназначен для хранения данных, необходимых для обучения и тестирования распределенных моделей глубокого обучения. Поскольку NoSQL представляет собой категорию баз данных, предназначенных для обработки больших данных. Мы приняли Apache Cassandra, распределенную систему управления базами данных NoSQL, первоначально разработанную Facebook. Она предназначена для управления большими объемами данных, распределенными по многим узлам, возможно, различным центрам обработки данных. Она предлагает несколько преимуществ, включая масштабируемость, высокую доступность, отказоустойчивость без ущерба для производительности (Cassandra, 2019; Lakshman & Malik, 2010).
<i>4.1.4. Big data processing layer</i>	4.1.4. Слой обработки больших данных
The main objective of this layer is to process the collected data from the previous layer, perform the pre-processing tasks such as data cleaning, tokenization, and converting words into the corresponding vector representations.	Основная задача этого уровня - обработка собранных данных с предыдущего уровня, выполнение задач предварительной обработки, таких как очистка данных, токенизация и преобразование слов в соответствующие векторные представления.
This layer relies on Apache Spark for performing large-scale data processing as it is a fast, an in-memory cluster computing framework. It has emerged as a strong successor to Apache Hadoop, essentially due to its higher capabilities for richer and faster analysis. Spark can run programs up 10x faster than Apache Hadoop on disk, or 100x faster in-memory (Ait Hammou, Ait Lahcen, & Mouline, 2018; Spark, 2019).	Этот уровень опирается на Apache Spark для выполнения крупномасштабной обработки данных, поскольку это быстрая среда кластерного вычисления в памяти. Он стал мощным преемником Apache Hadoop, в основном благодаря его более широким возможностям для более богатого и быстрого анализа. Spark может запускать программы в 10 раз быстрее, чем Apache Hadoop на диске, или в 100 раз быстрее в памяти (Ait Hammou, Ait Lahcen, & Mouline, 2018; Spark, 2019).

<p>In particular, Apache Spark is based on the so-called Resilient Distributed Dataset (RDD), which stands for Spark's fundamental abstraction. It is a read-only collection of partitioned data, which allows effective data reuse in a wide range of applications. Generally, RDD is a fault-tolerant data structure. It presents several advantages, such as optimizing distributed data placement, and the manipulation by employing a set of operators (Ait Hammou et al., 2018; Spark, 2019; Zaharia et al., 2012; Zaharia, Chowdhury, Franklin, Shenker, & Stoica, 2010).</p>	<p>В частности, Apache Spark основан на так называемом Resilient Distributed Dataset (RDD), который обозначает фундаментальную абстракцию Spark. Это доступная только для чтения коллекция разделенных данных, которая позволяет эффективно использовать данные в самых разных приложениях. Как правило, RDD является отказоустойчивой структурой данных. Он обладает рядом преимуществ, таких как оптимизация размещения распределенных данных и манипулирование с помощью набора операторов (Ait Hammou et al., 2018; Spark, 2019; Zaharia et al., 2012; Zaharia, Chowdhury, Franklin, Shenker, & Stoica, 2010).</p>
<p><i>4.1.5. Distributed deep learning for natural language processing layer</i></p>	<p>4.1.5. Распределенное глубокое обучение для слоя обработки естественного языка</p>
<p>This layer is devoted to training distributed deep learning models on large amounts of data spread out across several nodes. In particular, we have adopted three distributed Recurrent neural network variants for natural language processing, namely, Distributed Long Short-Term Memory (LSTM), Distributed Bidirectional Long Short-Term Memory (BiLSTM) and Distributed Gated Recurrent Unit (GRU).</p>	<p>Этот уровень посвящен обучению распределенных моделей глубокого обучения большим объемам данных, распределенным по нескольким узлам. В частности, мы приняли три варианта распределенной рекуррентной нейронной сети для обработки естественного языка, а именно: распределенную долговременную кратковременную память (LSTM), распределенную двунаправленную кратковременную память (BiLSTM) и распределенный управляемый рекуррентный блок (GRU).</p>
<p>In order to improve the performance of the aforementioned models in terms of classification accuracy, this step aims to learn the appropriate representation with respect to each review in the data set. We have employed BigDL framework, a deep learning library designed mainly for training large-scale distributed deep learning. It is developed by Intel for Apache Spark. BigDL enables writing distributed deep learning for big data applications as standard Apache Spark programs and running on a cluster of servers (BigDL, 2019; Wang et al., 2018b).</p>	<p>Чтобы повысить производительность вышеупомянутых моделей с точки зрения точности классификации, этот шаг направлен на изучение соответствующего представления в отношении каждого обзора в наборе данных. Мы использовали среду BigDL, библиотеку глубокого обучения, предназначенную главным образом для обучения крупномасштабному распределенному глубокому обучению. Он разработан Intel для Apache Spark. BigDL позволяет писать распределенное глубокое обучение для приложений с большими данными как стандартные программы Apache Spark и работать на кластере серверов (BigDL, 2019; Wang et al., 2018b).</p>
<p><i>4.1.6. Distributed machine learning layer</i></p>	<p>4.1.6. Распределенный слой машинного обучения</p>
<p>This step is dedicated to classify textual data, by exploiting the optimal representations generated by our proposal. In order to perform this task in the context of Big data, we have chosen a Spark's machine learning library called MLlib, which aims to make the development of large-scale machine learning</p>	<p>Этот шаг предназначен для классификации текстовых данных, используя оптимальные представления, сгенерированные нашим предложением. Чтобы выполнить эту задачу в контексте больших данных, мы выбрали библиотеку машинного обучения Spark под</p>

<p>applications scalable and easy to use. The MLlib provides a variety of fast and scalable implementations of machine learning algorithms for various tasks, such as dimensionality reduction, regression, clustering, and classification. It supports several languages, including Java, Python, Scala and R (Meng et al., 2016; MLlib, 2019)</p>	<p>названием MLlib, цель которой - сделать масштабные и простые в использовании приложения для машинного обучения крупномасштабными. MLlib предоставляет множество быстрых и масштабируемых реализаций алгоритмов машинного обучения для различных задач, таких как уменьшение размерности, регрессия, кластеризация и классификация. Он поддерживает несколько языков, включая Java, Python, Scala и R (Meng et al., 2016; MLlib, 2019)</p>
<p><i>4.1.7. Big data indexing layer</i></p>	<p>4.1.7. Слой индексации больших данных</p>
<p>This layer is intended to index data for providing efficient data storage, real-time search, and analyzing large amounts of data quickly. Therefore, in order to power our pipeline architecture by these functionalities, Elasticsearch is utilized as the underlying engine at this stage, which is a highly scalable framework built on top of Apache Lucene, a full-text search-engine library. It is a distributed search and analytics engine, where all fields in each stored document are indexed and searchable. Elasticsearch is primarily dedicated to supporting not only the search functionality but also complex aggregations (Elasticsearch, 2019; Gormley & Tong, 2015).</p>	<p>Этот уровень предназначен для индексации данных для обеспечения эффективного хранения данных, поиска в реальном времени и быстрого анализа больших объемов данных. Поэтому, для обеспечения нашей конвейерной архитектуры этими функциями, Elasticsearch используется в качестве базового механизма на данном этапе, который представляет собой масштабируемую среду, построенную на основе Apache Lucene, библиотеки полнотекстового поискового движка. Это распределенный механизм поиска и аналитики, в котором все поля в каждом сохраненном документе индексируются и доступны для поиска. Elasticsearch в первую очередь предназначен для поддержки не только функциональности поиска, но и сложных агрегатов (Elasticsearch, 2019; Gormley & Tong, 2015).</p>
<p><i>4.1.8. Real-time visualization layer</i></p>	<p>4.1.8. Слой визуализации в реальном времени</p>
<p>In order to explore and understand efficiently the data indexed in the previous stage, the current step is dedicated to real-time big data visualization. Particularly, it aims to perform big data analysis in a more comprehensible way, for gaining useful insights and extracting relevant information needed for serving specific objectives of several companies, like, business intelligence, real-time analysis of customer behavior and public opinion, and so on. Therefore, this step is based on a powerful platform called Kibana, which is devoted to visualizing data stored in Elasticsearch indexes. It exploits the powerful search and indexing functionalities of Elasticsearch framework available via its RESTful API to display several graphics. Kibana provides flexible analytics and visualization to facilitate the task of understanding large volumes of data (Kibana, 2019).</p>	<p>Для эффективного изучения и анализа данных, проиндексированных на предыдущем этапе, текущий этап посвящен визуализации больших данных в режиме реального времени. В частности, он нацелен на выполнение анализа больших данных более понятным способом, для получения полезной информации и извлечения соответствующей информации, необходимой для решения конкретных задач нескольких компаний, таких как бизнес-аналитика, анализ поведения потребителей и общественного мнения в режиме реального времени и т. Д. на. Поэтому этот шаг основан на мощной платформе Kibana, которая предназначена для визуализации данных, хранящихся в индексах Elasticsearch. Он</p>

	использует мощные функции поиска и индексирования инфраструктуры Elasticsearch, доступные через его RESTful API, для отображения нескольких графических объектов. Кибана обеспечивает гибкую аналитику и визуализацию, чтобы облегчить задачу понимания больших объемов данных (Кибана, 2019).
4.2. <i>Proposed method for sentiment analysis</i>	4.2. Предлагаемый метод анализа настроений
This subsection depicts our proposed technique for improving the performance of RNN variants. Fig. 2 illustrates the different steps.	В этом подразделе описана предлагаемая нами методика улучшения характеристик вариантов RNN. На рис. 2 показаны различные этапы.
<p>Fig. 2. Flowchart of the proposed method.</p>  <pre> graph TD A[(Data set)] --> B[Text pre-processing] B --> C[Word embedding (FastText)] C --> D[Learning text representation based on distributed DL models (LSTM/BiLSTM/GRU)] D --> E[Text classification] </pre>	Рис. 2. Блок-схема предлагаемого способа.
4.2.1. <i>Text pre-processing</i>	4.2.1. Предварительная обработка текста
Text pre-processing is an essential stage for text analytics. It refers to the process of preparing the data for sentiment analysis. As the reviews are usually composed of noisy and poorly structured sentences, it is necessary to transform the content of each review into the appropriate format to reduce the negative impact of the noise and facilitate text analytics. Therefore, this step is intended to perform a series of pre-processing such as removing the numbers, URLs, hashtags, and so on.	Предварительная обработка текста является важным этапом для анализа текста. Это относится к процессу подготовки данных для анализа настроений. Поскольку обзоры обычно состоят из шумных и плохо структурированных предложений, необходимо преобразовать содержание каждого обзора в соответствующий формат, чтобы уменьшить негативное влияние шума и облегчить анализ текста. Следовательно, этот шаг предназначен для выполнения серии предварительной обработки, такой как удаление номеров, URL-адресов, хэштегов и т. Д.
4.2.2. <i>Embedding layer</i>	4.2.2. Встраиваемый слой
In this step, the key idea of the embedding layer is to learn representations of words. Let $R_i = (w_1, w_2, \dots, w_N)$ be an input sequence, which represents a review composed of N words after padding. In order to process the review $R(i)$ based on a recurrent neural network variant model, it is common to transform textual data into numerical vectors, where each vector capture hidden information about the language. Based on the pre-trained fastText word representations (FastText, 2019), each review $R(i)$ in the data set can be converted into a sequence of vectors	На этом этапе ключевой идеей уровня внедрения является изучение представлений слов. Пусть $R_i = (w_1, w_2, \dots, w_N)$ - входная последовательность, представляющая обзор, состоящий из N слов после заполнения. Чтобы обработать обзор $R(i)$ на основе модели варианта рекуррентной нейронной сети, обычно преобразовывают текстовые данные в числовые векторы, где каждый вектор собирает скрытую информацию о языке. На основе

denoted as follows:	предварительно обученных представлений слова fastText (FastText, 2019) каждый обзор $R(i)$ в наборе данных может быть преобразован в последовательность векторов, обозначаемых следующим образом:
$V^{(i)} = \text{FastText}(R^{(i)}) = (v_1, \dots, v_N) \quad (12)$	
where each word w_i is represented by a D dimensional vector v_i .	где каждое слово w_i представлено D -мерным вектором v_i .
4.2.3. <i>Learning text representations based on distributed RNN variants</i>	4.2.3. Изучение текстовых представлений на основе распределенных вариантов RNN
The main goal of this step is to learn the optimal feature representations of each input sequence based on a more sophisticated procedure.	Основная цель этого шага состоит в том, чтобы узнать оптимальные представления характеристик каждой входной последовательности на основе более сложной процедуры.
Let $T = \{(x_i, y_i), i = 1, \dots, T \}$ be the training data set, which is composed of $ T $ instances. Each x_i and y_i stand for the sequence of word vectors and the label corresponding to a review $R(i)$, respectively. Also, let k be the parameter of representation, which determines the number of functions needed to infer the optimal representation. The first purpose is to take each input sequence of vectors $x_i = V(i)$ and convert it into an optimal vector representation $O(i)$ according to the following equation:	Пусть $T = \{(x_i, y_i), i = 1, \dots, T \}$ будет набор обучающих данных, который состоит из $ T $ экземпляров. Каждый x_i и y_i обозначает последовательность векторов слов и метку, соответствующую обзору $R(i)$ соответственно. Кроме того, пусть k будет параметром представления, который определяет количество функций, необходимых для определения оптимального представления. Первая цель состоит в том, чтобы взять каждую входную последовательность векторов $x_i = V(i)$ и преобразовать ее в оптимальное векторное представление $O(i)$ согласно следующему уравнению:
$O^{(i)} = g(V^{(i)}) = (o_1^{(i)}, \dots, o_k^{(i)}) \quad (13)$	
The key idea behind the function $g(.)$ is to consider the problem of learning representation of a sequence as a set of k subproblems, where each sub-problem can be resolved independently by minimizing a separate objective function. The element o_{i1} denotes the value estimated after the minimization process. The set of functions F is typically expressed as follows:	Основная идея функции $g(.)$ состоит в том, чтобы рассмотреть проблему изучения представления последовательности в виде набора из k подзадач, где каждая подзадача может быть решена независимо путем минимизации отдельной целевой функции. Элемент o_{i1} обозначает значение, оцененное после процесса минимизации. Набор функций F обычно выражается следующим образом:

$F = (F_j, j = 1, \dots, k)$ $F_j = -\frac{1}{ T } \sum_{i=1}^{ T } \sum_{c=1}^C y_i^{(c)} \log(\hat{p}_i^{(c)})$ <p style="text-align: center;">(14)</p>	
<p>where $y_i(c)$ equals 1 if the ground-truth label for the instance x_i is c, else, it is equal to 0. C is the number of output labels, and $\hat{p}_i(c)$ denotes the estimated probability for the label c. Each function F_j is a cross entropy loss function associated to a deep learning model, which is able to handle variable-length input sequences and learn long-term dependencies in sequential data.</p>	<p>где $y_i(c)$ равно 1, если метка истинности основания для экземпляра x_i равна c, в противном случае она равна 0. C - количество выходных меток, а $\hat{p}_i(c)$ обозначает предполагаемую вероятность для метки c. Каждая функция F_j является функцией кросс-энтропийной потери, связанной с моделью глубокого обучения, которая способна обрабатывать входные последовательности переменной длины и изучать долгосрочные зависимости в последовательных данных.</p>
<p>Therefore, each classification problem can be solved based on a distributed Recurrent neural network variant. Specifically, in this step, we have adopted three models, namely, Long short-term memory (LSTM), Bidirectional long short-term memory (BiLSTM) and Gated recurrent unit (GRU). Each model stacks multiple layers as defined below:</p>	<p>Следовательно, каждая проблема классификации может быть решена на основе распределенного варианта рекуррентной нейронной сети. В частности, на этом этапе мы приняли три модели, а именно: Долгосрочная кратковременная память (LSTM), Двухнаправленная долговременная кратковременная память (BiLSTM) и управляемый рекуррентный блок (GRU). Каждая модель укладывает несколько слоев, как определено ниже:</p>
<ul style="list-style-type: none"> • Input Layer: This first layer receives each review $R(i)$ as a sequence of 300-dimensional vectors $V(i)$. • Distributed Recurrent neural network variant Layer: This second layer employs a LSTM, BiLSTM or GRU model to process the input sequences and output the hidden state at the last time step denoted $h(i)$. • Dense Layer: The third one is a fully connected layer of 100 nodes, which processes the last vector state $h(i)$, and produces a hidden representation $dl(i)$ 100. Leaky ReLU is used as the nonlinear activation function with the parameter = 0.3 (Maas, Hannun, & Ng, 2013). Dropout (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012) is used to prevent the overfitting. • Softmax Layer: The softmax layer estimates the probability distribution vector $\hat{p}^{(i)} C$, which corresponds to the C target classes. 	<ul style="list-style-type: none"> • Входной уровень: этот первый слой получает каждый обзор $R(i)$ в виде последовательности 300-мерных векторов $V(i)$. • Уровень распределенной рекуррентной нейронной сети. Этот второй уровень использует модель LSTM, BiLSTM или GRU для обработки входных последовательностей и вывода скрытого состояния на последнем временном шаге, обозначенном $h(i)$. • Плотный слой. Третий - это полностью связанный слой из 100 узлов, который обрабатывает последнее векторное состояние $h(i)$ и создает скрытое представление $dl(i)$ 100. Leaky ReLU используется в качестве нелинейной функции активации с параметром = 0,3 (Maas, Hannun & Ng, 2013). Выпадение (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012) используется для предотвращения переобучения. • Слой Softmax: Слой Softmax оценивает вектор распределения вероятности $\hat{p}^{(i)} C$, который соответствует целевым классам C.

<p>To be more specific, Eqs. (15)–(17) present the underlying operations with respect to the models LSTM, BiLSTM and GRU, respectively.</p>	<p>Чтобы быть более конкретным, уравнения. (15) - (17) представляют базовые операции в отношении моделей LSTM, BiLSTM и GRU соответственно.</p>
$ \begin{aligned} h^{(i)} &= LSTM(V^{(i)}) \\ dl^{(i)} &= dense(h^{(i)}) \\ \hat{p}^{(i)} &= softmax(dl^{(i)}), \hat{p}^{(i)} \in \mathbb{R}^C \\ M_{LSTM}(V^{(i)}) &= \arg \max_{c \in C}(\hat{p}^{(i)}) \end{aligned} \tag{15} $	
$ \begin{aligned} h^{(i)} &= BiLSTM(V^{(i)}) \\ dl^{(i)} &= dense(h^{(i)}) \\ \hat{p}^{(i)} &= softmax(dl^{(i)}), \hat{p}^{(i)} \in \mathbb{R}^C \\ M_{BiLSTM}(V^{(i)}) &= \arg \max_{c \in C}(\hat{p}^{(i)}) \end{aligned} \tag{16} $	
$ \begin{aligned} h^{(i)} &= GRU(V^{(i)}) \\ dl^{(i)} &= dense(h^{(i)}) \\ \hat{p}^{(i)} &= softmax(dl^{(i)}), \hat{p}^{(i)} \in \mathbb{R}^C \\ M_{GRU}(V^{(i)}) &= \arg \max_{c \in C}(\hat{p}^{(i)}) \end{aligned} \tag{17} $	
<p>According to these models, we distinguish three different architectures denoted <i>ALSTM</i>, <i>ABiLSTM</i> and <i>AGRU</i>. Each one consists of a set of k models represented as follows:</p>	<p>В соответствии с этими моделями мы различаем три разные архитектуры, обозначаемые ALSTM, ABiLSTM и AGRU. Каждая состоит из набора из k моделей, представленных следующим образом:</p>
$A_{LSTM} = (M_{LSTM}^{(1)}, \dots, M_{LSTM}^{(k)}) \tag{18}$	
$A_{BiLSTM} = (M_{BiLSTM}^{(1)}, \dots, M_{BiLSTM}^{(k)}) \tag{19}$	
$A_{GRU} = (M_{GRU}^{(1)}, \dots, M_{GRU}^{(k)}) \tag{20}$	

where $MLSTM$, $(j) M$, $BiLSTM$ $(j) MGRU$ (j) denotes the j th LSTM, BiLSTM, GRU model, respectively.	где $MLSTM$, $(j) M$, $BiLSTM$ $(j) MGRU$ (j) обозначает j -ую модель LSTM, BiLSTM, GRU соответственно.
The underlying rationale behind adopting one of the Eqs. (18)–(20) for representing sequential data is the assumption that, enhancing data representation plays a fundamental role for boosting the performance of classifiers, as well as overcoming the weaknesses of existing research works.	Основное обоснование принятия одного из уравнений. (18) - (20) для представления последовательных данных является предположение, что улучшение представления данных играет фундаментальную роль для повышения производительности классификаторов, а также преодоления слабых сторон существующих исследовательских работ.
In particular, after training the set of k distributed models of an architecture $ALSTM$, the values estimated by each model $MLSTM$ (j) are combined to produce an optimal representation. Thus, the input sequence $V(i)$ can be re-represented as follows:	В частности, после обучения набора из k распределенных моделей архитектуры $ALSTM$ значения, оцениваемые каждой моделью $MLSTM$ (j) , объединяются для получения оптимального представления. Таким образом, входная последовательность V (i) может быть представлена следующим образом:
$O^{(i)} = (M_{LSTM}^{(1)}(V^{(i)}), \dots, M_{LSTM}^{(k)}(V^{(i)})) \quad (21)$	
where $MLSTM$ (V) , $(1) (i)$ and $O(i) k$ stands for the new representation for the review $R(i)$.	где $MLSTM$ (V) , $(1) (i)$ и $O(i) k$ обозначает новое представление для обзора $R(i)$.
Similarly, in case of using an ensemble $ABiLSTM$ or $AGRU$, the optimal representations are formulated as follows:	Аналогично, в случае использования ансамбля $ABiLSTM$ или $AGRU$, оптимальные представления формулируются следующим образом:
$O^{(i)} = (M_{BiLSTM}^{(1)}(V^{(i)}), \dots, M_{BiLSTM}^{(k)}(V^{(i)})) \quad (22)$	
$O^{(i)} = (M_{GRU}^{(1)}(V^{(i)}), \dots, M_{GRU}^{(k)}(V^{(i)})) \quad (23)$	
4.2.4. Text classification	4.2.4. Классификация текста
In general, text classification is defined as the problem of learning classification model from reviews with pre-defined labels. Then, the learned model is employed to classify reviews in the future.	В целом, классификация текста определяется как проблема изучения модели классификации на основе обзоров с заранее определенными метками. Затем изученная модель используется для классификации обзоров в будущем.
Logistic Regression is one of the well-known machine learning algorithms. It is a supervised learning method, which is commonly used for classification tasks. In this step, after the representation process, the training set can be represented as follows:	Логистическая регрессия является одним из известных алгоритмов машинного обучения. Это контролируемый метод обучения, который обычно используется для задач классификации. На этом этапе после процесса представления обучающий набор может быть представлен следующим образом:
$T = \{(x_i, y_i), i = 1, \dots, T \}, x_i \in \mathbb{R}^k \quad (24)$	

where $x_i = O(i)$ stands for the estimated representation (i.e., the features), and y_i is the class label associated to the review $R(i)$.	где $x_i = O(i)$ обозначает предполагаемое представление (то есть признаки), а y_i обозначает метку класса, связанную с обзором $R(i)$.
The main objective is to train a multinomial logistic regression model that can take a vector $O(i)$ k as input, and predict the probabilities of the different possible classes.	Основная цель состоит в том, чтобы обучить модель многочленной логистической регрессии, которая может принимать вектор $O(i)$ k в качестве входных данных, и прогнозировать вероятности различных возможных классов.
Using the trained model, the probability that x_i belongs to the class label y_i is expressed as follows (Cawley, Talbot, & Girolami, 2007; Krishnapuram, Carin, Figueiredo, & Hartemink, 2005):	Используя обученную модель, вероятность того, что x_i принадлежит метке класса y_i , выражается следующим образом (Cawley, Talbot, & Girolami, 2007; Krishnapuram, Carin, Figueiredo, & Hartemink, 2005):
$P(y_i^{(c)} = 1 x_i; \theta) = \frac{\exp(\theta^{(c)T} x_i)}{\sum_{j=1}^C \exp(\theta^{(j)T} x_i)} \quad (25)$	
where $c \in \{1, \dots, C\}$, $y_i(c)$ equals 1 if the class label is c , otherwise, it is equal to 0. $\theta^{(c)}$ denotes the parameter vector corresponding to the class c . T denotes the matrix transpose, and θ represents all the parameters of the model.	где $c \in \{1, \dots, C\}$, $y_i(c)$ равно 1, если метка класса равна c , в противном случае она равна 0. $\theta^{(c)}$ обозначает вектор параметров, соответствующий классу c . T обозначает транспонирование матрицы, а θ представляет все параметры модели.
Finally, based on the estimated probabilities, the model generates the predicted class.	Наконец, на основе оцененных вероятностей модель генерирует предсказанный класс.
5. Experiments	5. Эксперименты
5.1. Data sets	5.1. Наборы данных
In this work, the experiments are conducted using two real-world data sets, namely, Yelp and Sentiment140.	В этой работе эксперименты проводятся с использованием двух реальных наборов данных, а именно Yelp и Sentiment140.
<ul style="list-style-type: none"> • Yelp: This data set is composed 6,685,900 classified reviews provided by 1,637,138 users for 192,609 businesses. In this work, we have randomly selected 100,000 reviews as the original data set. We have considered 1 and 2 stars as a negative class, while 4 and 5 as a positive class (Yelp, 2019). • Sentiment140: This is a Twitter sentiment analysis data set, which is originated from Stanford University. This particular data set consists of 1,600,000 classified tweets. In this work, we have randomly selected 20,000 tweets as the original data set (Sentiment140, 2019). 	<ul style="list-style-type: none"> • Yelp: Этот набор данных состоит из 6 685 900 классифицированных обзоров, предоставленных 1 637 138 пользователями для 192 609 предприятий. В этой работе мы случайным образом выбрали 100 000 обзоров в качестве исходного набора данных. Мы рассматривали 1 и 2 звезды как отрицательный класс, а 4 и 5 как положительный класс (Yelp, 2019). • Sentiment140: это набор данных для анализа настроений Twitter, созданный в Стэнфордском университете. Этот конкретный набор данных состоит из 1 600 000 классифицированных твитов. В этой работе мы случайным образом выбрали 20 000 твитов в качестве исходного набора данных (Sentiment140, 2019).

5.2. <i>Evaluation metrics</i>	5.2. Метрики оценки
To compare the performance of classifiers, we have adopted two metrics, namely, accuracy and F-score. They are widely used to assess the performance of sentiment analysis systems (Abdi, Shamsuddin, Hasan, & Piran, 2019; Al-Smadi, Al-Ayyoub, Jararweh, & Qawasmeh, 2019). The Accuracy and F-score metrics are defined as follows:	Чтобы сравнить производительность классификаторов, мы приняли две метрики, а именно точность и F-показатель. Они широко используются для оценки эффективности систем анализа настроений (Абди, Шамсуддин, Хасан и Пирани, 2019; Аль-Смади, Аль-Айюб, Джарарве и Кавасмех, 2019). Метрики точности и F-показателя определяются следующим образом:
$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (26)$	
$Precision = \frac{TP}{TP + FP}$ $Recall = \frac{TP}{TP + FN}$ $F - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (27)$	
<p>Where</p> <ul style="list-style-type: none"> • TN is the number of true positives. • TP refers to the number of true positives. • FP denotes the number of false positives • FN is the number of false negatives. 	<p>где</p> <ul style="list-style-type: none"> • TN - количество истинных позитивов. • TP относится к числу истинных позитивов. • FP обозначает количество ложных срабатываний. • FN - количество ложных срабатываний.
5.3. <i>Methodology</i>	5.3. методология
For the experimental study, each data set is partitioned into two parts. The training set is composed of 90% of data, while the test set contains 10%. The training set is used for training the model, and the test set for assessing the performance. This procedure is repeated 5 times. Then the average is computed.	Для экспериментального исследования каждый набор данных разделен на две части. Учебный набор состоит из 90% данных, в то время как тестовый набор содержит 10%. Учебный комплект используется для обучения модели, а тестовый набор для оценки производительности. Эта процедура повторяется 5 раз. Затем вычисляется среднее значение.
5.4. <i>Implementation detail</i>	5.4. Детали реализации
The experiments have been executed on a cluster, which is composed of two nodes: one master node, and a slave node.	Эксперименты были выполнены на кластере, который состоит из двух узлов: одного главного узла и подчиненного узла.
In addition, we have used fastText1 for distributed word representations. The deep learning models were written in Python. Details of the various frameworks adopted in this work are as follows:	Кроме того, мы использовали fastText1 для распределенных представлений слов. Модели глубокого обучения были написаны на Python. Детали различных структур, принятых в этой работе,

	следующие:
PySpark 2.3.3 • Cassandra 3.0.15 • Kafka 2.1.0 • BigDL 0.7.0 • Spark MLLib • Kibana 6.6.2 • Elasticsearch 6.6.2 • Python 3.5 • Ubuntu 16.04 LTS	
<i>5.5. Parameter setting</i>	5.5. Установка параметра
An extensive number of experimentations were conducted to determine the best settings. The parameters used for each method are presented as follows.	Было проведено большое количество экспериментов для определения наилучших настроек. Параметры, используемые для каждого метода, представлены следующим образом.
In general, the different deep learning models were optimized using Adam optimization algorithm (Kingma & Ba, 2015). The batch size was set to 64. The dropout (Hinton et al., 2012) rate for the dense layer was fixed at 0.5, while the recurrent dropout was set to 0.2. The parameter of representation $k = 9$. The learning rate was fixed at 0.01.	В целом, различные модели глубокого обучения были оптимизированы с использованием алгоритма оптимизации Адама (Kingma & Ba, 2015). Размер партии был установлен равным 64. Частота выпадения (Hinton et al., 2012) для плотного слоя была установлена равной 0,5, в то время как периодическое выпадение было установлено равной 0,2. Параметр представления $k = 9$. Скорость обучения была зафиксирована на уровне 0,01.
<i>5.6. Experimental results</i>	5.6. Результаты экспериментов
This section presents the experiments carried out using two real-world data sets. Particularly, our proposal is compared with the following deep learning approaches:	В этом разделе представлены эксперименты, проведенные с использованием двух реальных наборов данных. В частности, наше предложение сравнивается со следующими подходами глубокого обучения:
<ul style="list-style-type: none"> • LSTM: Long short-term memory model with the fastText word representations. • BiLSTM (Graves et al., 2013): Bidirectional long short-term memory model with the pre-trained fastText word representations. • GRU (Cho et al., 2014; Chung et al., 2014): Gated recurrent unit model with the pre-trained fastText word representations. • XGBoost-avg-fastText (Chen & Guestrin, 2016): Extreme gradient boosting algorithm based on the average of fastText word embeddings. • RF-avg-fastText: Random forest classifier based on the average of fastText word embeddings. • Our proposal-LSTM: The proposed distributed method based on LSTM layer and fastText word representations. • Our proposal-BiLSTM: The proposed distributed method based on BiLSTM layer and fastText word representations. • Our proposal-GRU: The proposed distributed method based on GRU layer and fastText word representations. 	<ul style="list-style-type: none"> • LSTM: модель долговременной кратковременной памяти с представлениями слов fastText. • BiLSTM (Graves et al., 2013): двунаправленная модель долгосрочной кратковременной памяти с предварительно обученными представлениями слов fastText. • GRU (Cho et al., 2014; Chung et al., 2014): модель стробируемого рекуррентного блока с предварительно обученными представлениями слова fastText. • XGBoost-avg-fastText (Chen & Guestrin, 2016): алгоритм экстремального повышения градиента, основанный на среднем значении вложений слов fastText. • RF-avg-fastText: классификатор случайных лесов, основанный на среднем значении вложений слов fastText. • Наше предложение - LSTM: предлагаемый распределенный метод, основанный на представлениях слоя LSTM и fastText. • Наше предложение BiLSTM: предлагаемый распределенный метод, основанный на представлении слоев BiLSTM и текстовых

слов fastText. • Наше предложение - GRU: предлагаемый распределенный метод, основанный на представлениях уровня GRU и слов fastText.

Tables 1 and 2 present the experimental results on the sentiment140 and Yelp data sets, respectively. Each table reports the comparative results in terms of accuracy and F-score of our model against well-known deep learning approaches used for sentiment analysis task. To highlight the best result, we mark in bold the best score.

В таблицах 1 и 2 представлены экспериментальные результаты для наборов данных sentiment140 и Yelp соответственно. В каждой таблице представлены сравнительные результаты с точки зрения точности и F-показателя нашей модели по сравнению с хорошо известными подходами глубокого обучения, используемыми для задачи анализа настроений. Чтобы подчеркнуть лучший результат, мы отмечаем жирным шрифтом лучший результат.

Table 1 Experimental results on sentiment140 data set (average for 5 runs).

Model	Accuracy	<i>F</i> -score
RF-avg-fastText	0.7265	0.7227
XGBoost-avg-fastText	0.736	0.7297
LSTM	0.7745	0.7777
BiLSTM	0.7792	0.7771
GRU	0.7826	0.7882
Our proposal-LSTM	0.7809	0.7843
Our proposal-BiLSTM	0.7876	0.7865
Our proposal-GRU	0.7888	0.7902

Таблица 1 Экспериментальные результаты для набора данных sentiment140 (в среднем за 5 прогонов).

Table 2 Experimental results on Yelp data set (average for 5 runs).

Model	Accuracy	<i>F</i> -score
RF-avg-fastText	0.8453	0.8464
XGBoost-avg-fastText	0.856	0.8562
LSTM	0.9174	0.9191
BiLSTM	0.9229	0.9237
GRU	0.9239	0.9219
Our proposal-LSTM	0.9255	0.9269
Our proposal-BiLSTM	0.9291	0.9295
Our proposal-GRU	0.9328	0.9310

Таблица 2 Экспериментальные результаты на наборе данных Yelp (в среднем за 5 прогонов).

From Tables 1 and 2, it is obvious that our proposal shows higher accuracy and *F*-score than other state-of-the-art methods. It substantially outperforms the other competitors on both data sets. In particular, compared with the recurrent neural network variants, Distributed Long short-term memory (LSTM), Distributed Bidirectional long short-term memory (BiLSTM) and Distributed Gated recurrent

Из таблиц 1 и 2 очевидно, что наше предложение показывает более высокую точность и F-показатель, чем другие современные методы. Это существенно превосходит других конкурентов в обоих наборах данных. В частности, по сравнению с рекуррентными вариантами нейронной сети, распределенной

unit (GRU) models, the obtained results demonstrate that the improved methods (i.e., Our proposal-LSTM, Our proposal-BiLSTM, Our proposal-GRU) yield higher accuracy and <i>F</i> -score in comparison with the original methods.	долговременной кратковременной памятью (LSTM), распределенной двунаправленной кратковременной памятью (BiLSTM) и распределенными рекуррентными единицами Gated, полученные результаты показывают, что улучшенные методы (т.е. наше предложение-LSTM, наше предложение-BiLSTM, наше предложение-ГРУ) дают более высокую точность и F-оценку по сравнению с оригинальными методами.
In view of the obtained results, it is reasonable to conclude that our proposal is able to enhance the performance of several existing works and thus contribute to more efficient models for big data sentiment analysis.	С учетом полученных результатов разумно сделать вывод, что наше предложение способно повысить производительность нескольких существующих работ и, таким образом, внести вклад в более эффективные модели для анализа настроений больших данных.
6. Conclusions	6. Выводы
With the advent of social media applications, social big data has become an important topic for a large number of research areas. In this paper, we have proposed a distributed intelligent system for real-time social big data analytics. It is primarily designed based on a set of powerful big data tools to manage and process, and analyze large-scale data efficiently, as well as, improve real-time decision-making processes. In addition, we have devised an effective solution to improve the performance of a set of distributed deep learning models with fastText for sentiment analysis.	С появлением приложений для социальных сетей большие социальные данные стали важной темой для большого числа областей исследований. В этой статье мы предложили распределенную интеллектуальную систему для анализа социальных данных в реальном времени. Он в первую очередь разработан на основе набора мощных инструментов для работы с большими данными, которые позволяют эффективно управлять и обрабатывать большие объемы данных, а также улучшать процессы принятия решений в реальном времени. Кроме того, мы разработали эффективное решение для повышения производительности набора распределенных моделей глубокого обучения с помощью fastText для анализа настроений.
Extensive experiments demonstrate the promise of our work. In particular, the results show that our proposed solution is able to increase the accuracy of well-known deep learning models (i.e., Long short-term memory (LSTM), Bidirectional long short-term memory (BiLSTM) and Gated recurrent unit (GRU)).	Обширные эксперименты демонстрируют перспективность нашей работы. В частности, результаты показывают, что наше предлагаемое решение способно повысить точность хорошо известных моделей глубокого обучения (например, Долгосрочная кратковременная память (LSTM), Двунаправленная долговременная кратковременная память (BiLSTM) и
Furthermore, the current work can provide many benefits for practitioners and researchers who want to collect, handle, analyze and visualize several sources of information in real-time. Also, it contributes to a better understanding of public opinion and user behavior, thanks to the improved variants of the most powerful distributed deep learning and machine learning algorithms. Moreover, it allows	Кроме того, текущая работа может предоставить множество преимуществ для практиков и исследователей, которые хотят собирать, обрабатывать, анализировать и визуализировать несколько источников информации в режиме реального времени. Кроме того, это способствует лучшему пониманию

improving several existing works based on RNN variants for natural language processing.	общественного мнения и поведения пользователей благодаря улучшенным вариантам самых мощных распределенных алгоритмов глубокого обучения и машинного обучения. Кроме того, это позволяет улучшить несколько существующих работ на основе вариантов RNN для обработки естественного языка.
For future work, we plan to investigate the impact of incorporating additional information into fastText vector representations for solving the sentiment analysis problem in the context of big data. As a second future work, it would be interesting to develop other strategies to further improve the performance of distributed deep learning models for natural language processing.	Для дальнейшей работы мы планируем изучить влияние включения дополнительной информации в векторные представления fastText для решения проблемы анализа настроений в контексте больших данных. В качестве второй будущей работы было бы интересно разработать другие стратегии для дальнейшего повышения производительности распределенных моделей глубокого обучения для обработки естественного языка.
Acknowledgments	
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.	
<p>Abdi, A., Shamsuddin, S. M., Hasan, S., & Piran, J. (2019). Deep learning-based sentiment classification of evaluative text based on multi-feature fusion. <i>Information Processing & Management</i>, 56(4), 1245–1259.</p> <p>Ait Hammou, B., Ait Lahcen, A., & Mouline, S. (2018). Apra: An approximate parallel recommendation algorithm for big data. <i>Knowledge-Based Systems</i>, 157, 10–19.</p> <p>Al-Smadi, M., Al-Ayyoub, M., Jararweh, Y., & Qawasmeh, O. (2019). Enhancing aspect-based sentiment analysis of arabic hotels' reviews using morphological, syntactic and semantic features. <i>Information Processing & Management</i>, 56(2), 308–319.</p> <p>Alharbi, A. S. M., & de Doncker, E. (2019). Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. <i>Cognitive Systems Research</i>, 54, 50–61.</p> <p>Bello-Orgaz, G., Jung, J. J., & Camacho, D. (2016). Social big data: Recent achievements and new challenges. <i>Information Fusion</i>, 28, 45–59.</p> <p>BigDL (2019). Distributed deep learning library for apache spark. (Accessed: 10 April 2019) URL: https://software.intel.com/en-us/ai/frameworks/bigdl.</p>	

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.

Cassandra (2019). Apache cassandra. (Accessed: 10 April 2019) URL: <http://cassandra.apache.org/>.

Cawley, G. C., Talbot, N. L., & Girolami, M. (2007). *Sparse multinomial logistic regression via Bayesian l1 regularisation*. *Advances in neural information processing systems* 209–216.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2), 171–209. <https://doi.org/10.1007/s11036-013-0489-0>.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system*. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM 785–794.

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, 72, 221–230.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., & Schwenk, H. (2014). *Learning phrase representations using RNN encoder–decoder for statistical machine translation*. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics 1724–1734. <https://doi.org/10.3115/v1/D14-1179>.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. *NIPS 2014 workshop on deep learning*, December 2014.

Del Vecchio, P., Mele, G., Ndou, V., & Secundo, G. (2018). Creating value from social big data: implications for smart tourism destinations. *Information Processing & Management*, 54(5), 847–860.

Elasticsearch (2019). Elasticsearch. (Accessed: 10 April 2019) URL:

<https://www.elastic.co/products/elasticsearch>.

FastText (2019). Fasttext: Library for efficient text classification and representation learning. (Accessed: 10 April 2019) URL: <https://fasttext.cc/>.

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), 137–144.

García-Gil, D., Luengo, J., García, S., & Herrera, F. (2019). Enabling smart data: Noise filtering in big data classification. *Information Sciences*, 479, 135–152.

García-Gil, D., Ramírez-Gallego, S., García, S., & Herrera, F. (2017). A comparison on scalability for batch big data processing on apache spark and apache flink. *Big Data Analytics*, 2(1), 1.

Gormley, C., & Tong, Z. (2015). *Elasticsearch: The definitive guide: A distributed real-time search and analytics engine*. O'Reilly Media, Inc.

Graves, A., Mohamed, A., & Hinton, G. (2013). *Speech recognition with deep recurrent neural networks. 2013 IEEE international conference on acoustics, speech and signal processing*6645–6649. <https://doi.org/10.1109/ICASSP.2013.6638947>.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5–6), 602–610.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Hu, H., Wen, Y., Chua, T.-S., & Li, X. (2014). Toward scalable systems for big data analytics: A technology tutorial. *IEEE Access*, 2, 652–687.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., & Daumé III, H. (2015). *Deep*

unordered composition rivals syntactic methods for text classification. Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)1.
Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers) 1681–1691.

Jan, B., Farman, H., Khan, M., Imran, M., Islam, I. U., Ahmad, A., et al. (2019). Deep learning in big data analytics: A comparative study. *Computers & Electrical Engineering*, 75, 275–287.

Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, 5, 2870–2879.

Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, 6, 23253–23260.

Jimenez-Marquez, J. L., Gonzalez-Carrasco, I., Lopez-Cuadrado, J. L., & Ruiz-Mezcua, B. (2019). Towards a big data framework for analyzing social media content. *International Journal of Information Management*, 44, 1–12.

Kafka (2019). Apache kafka. (Accessed: 10 April 2019). URL:<https://kafka.apache.org/>.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). *A convolutional neural network for modelling sentences. Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)*. Baltimore, Maryland: Association for Computational Linguistics 655–665. <https://doi.org/10.3115/v1/P14-1062>.

Kibana (2019). Kibana. (Accessed: 10 April 2019). URL:<https://www.elastic.co/products/kibana>.

Kim, Y. (2014). *Convolutional neural networks for sentence classification. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics 1746–1751.

<https://doi.org/10.3115/v1/D14-1181>.

Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. (2016). *Character-aware neural language models. Thirtieth AAAI conference on artificial intelligence*.

Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization. 3rd international conference on learning representations (ICLR)*.

Kreps, J., Narkhede, N., Rao, J., et al. (2011). *Kafka: A distributed messaging system for log processing. Proceedings of the NETDB1–7*.

Krishnapuram, B., Carin, L., Figueiredo, M. A., & Hartemink, A. J. (2005). Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6), 957–968.

Lakshman, A., & Malik, P. (2010). Cassandra: A decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, 44(2), 35–40.

Lee, I. (2017). Big data: Dimensions, evolution, impacts, and challenges. *Business Horizons*, 60(3), 293–303.

Lin, B. Y., Xu, F., Luo, Z., & Zhu, K. (2017). *Multi-channel BILSTM-CRF model for emerging named entity recognition in social media. Proceedings of the 3rd workshop on noisy user-generated text*160–165.

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337, 325–338.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). *Rectifier nonlinearities improve neural network acoustic models. Proceedings of the 30th international conference on machine learning (ICML-13)*.

McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D., & Barton, D. (2012). Big data: the management revolution. *Harvard Business Review*, 90(10), 60–68.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., et al. (2016). Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research*, 17(1), 1235–1241.

Mikolov, T., Grave, E., Bojanowski, P., Puhersch, C., & Joulin, A. (2018). *Advances in pre-training distributed word representations. Proceedings of the international conference on language resources and evaluation (LREC 2018)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). *Distributed*

representations of words and phrases and their compositionality. *Proceedings of the 26th international conference on neural information processing systems – volume 2NIPS’13* 3111–3119.

MLlib (2019). Machine learning library (mllib). (Accessed: 10 April 2019). URL: <https://spark.apache.org>.

Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, 20(4), 2923–2960.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1), 1.

Ngai, E. W., Tao, S. S., & Moon, K. K. (2015). Social media research: Theories, constructs, and conceptual frameworks. *International Journal of Information Management*, 35(1), 33–44.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* 1532–1543.

Pham, D.-H., & Le, A.-C. (2018). Exploiting multiple word embeddings and one-hot character vectors for aspect-based sentiment analysis. *International Journal of Approximate Reasoning*, 103, 1–10.

Ragini, J. R., Anand, P. R., & Bhaskar, V. (2018). Big data analytics for disaster response and recovery through sentiment analysis. *International Journal of Information Management*, 42, 13–24.

Rezaeinia, S. M., Rahmani, R., Ghodsi, A., & Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117, 139–147.

Saggi, M. K., & Jain, S. (2018). A survey towards an integration of big data analytics to big insights for value-creation. *Information Processing &*

Management, 54(5), 758–790.

Salehan, M., & Kim, D. J. (2016). Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics. *Decision Support Systems*, 81, 30–40.

Sentiment140 (2019) data set. (Accessed: 2 March 2019) URL: <http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip>.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). *Semi-supervised recursive autoencoders for predicting sentiment distributions. Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics 151–161.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). *Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of the 2013 conference on empirical methods in natural language processing* 1631–1642.

Song, M., Park, H., & Shin, K.-S. (2019). Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in korean. *Information Processing & Management*, 56(3), 637–653.

Spark (2019). Apache spark. (Accessed: 10 April 2019) URL: <https://spark.apache.org/>.

Stieglitz, S., Mirbabaie, M., Ross, B., & Neuberger, C. (2018). Social media analytics—challenges in topic discovery, data collection, and data preparation. *International Journal of Information Management*, 39, 156–168.

Sutskever, I., Martens, J., & Hinton, G. E. (2011). *Generating text with recurrent neural networks. Proceedings of the 28th international conference on machine learning (ICML-11)* 1017–1024.

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks,. (pp. 1556–1566) URL: <https://www.aclweb.org/anthology/P15-1150.10.3115/v1/P15-1150>.

Tien, N. H., & Le, N. M. (2017). *An ensemble method with sentiment features and clustering support. Proceedings of the eighth international joint conference on natural language processing (volume 1: Long papers)1. Proceedings of the eighth international joint conference on natural language processing (volume 1: Long papers)* 644–653.

Valdivia, A., Luzyn, M. V., & Herrera, F. (2017). Sentiment analysis in tripadvisor. *IEEE Intelligent Systems*, 32(4), 72–77.

Vosoughi, S., Vijayaraghavan, P., & Roy, D. (2016). *Tweet2vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder. Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval.* ACM1041–1044.

Wang, J., Yu, L.-C., Lai, K. R., & Zhang, X. (2016). *Dimensional sentiment analysis using a regional CNN-LSTM model. Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)2. Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* 225–230.

Wang, X., Liu, Y., Chengjie, S., Wang, B., & Wang, X. (2015). *Predicting polarities of tweets by composing word embeddings with long short-term memory. Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)1. Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* 1343–1353.

Wang, Y., Kung, L., & Byrd, T. A. (Kung, Byrd, 2018a). Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological Forecasting and Social Change*, 126, 3–13.

Wang, Y., Qiu, X., Ding, D., Zhang, Y., Wang, Y., Jia, X., et al. (2018). Bigdl: A distributed deep learning framework for big data. arXiv preprint arXiv:1804.05839.

Xiang, Z., Schwartz, Z., Gerdes Jr, J. H., & Uysal, M. (2015). What can big data

and text analytics tell us about hotel guest experience and satisfaction?
International Journal of Hospitality Management, 44, 120–130.

Yelp (2019). Yelp data set. (Accessed: 2 March 2019) URL: <https://www.yelp.com/dataset/challenge>.

Yu, L.-C., Wang, J., Lai, K. R., & Zhang, X. (2017). *Refining word embeddings for sentiment analysis. Proceedings of the 2017 conference on empirical methods in natural language processing* 534–539.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., et al. (2012). *Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. Proceedings of the 9th USENIX conference on networked systems design and implementation NSDI'12* Berkeley, CA, USA: USENIX Association.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). *Spark: Cluster computing with working sets. Proceedings of the 2nd unix conference on hot topics in cloud computing HotCloud'10* Berkeley, CA, USA: USENIX Association 10–10.

Zhang, Q., Yang, L. T., & Chen, Z. (2016). Deep computation model for unsupervised feature learning on big data. *IEEE Transactions on Services Computing*, 9(1), 161–171.

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, 42, 146–157.