УДК 004.89

КЛАССИФИКАТОР ТОНАЛЬНОСТИ ТЕКСТОВ НА ОСНОВЕ LSTM-НЕЙРОННОЙ СЕТИ

Н.М. Лыченко, А.В. Сороковая

Обосновано применение LSTM-сети для решения задачи классификации тональности текстов и на основе этой модели разработаны нейросетевой классификатор и программная система для классификации тональности текстов. Программная система представляет собой библиотеку программных модулей, отвечающих за алгоритмы считывания текстов, токенизации, векторного представления слов, классификации тональности текстов и оценки качества алгоритмов и Web API, который обслуживает запросы клиента с использованием разработанной библиотеки.

Ключевые слова: определение тональности текста; задача классификации; токенизация; нейронные сети; LSTM-сеть; библиотека программных модулей; WEB API.

LSTM-НЕЙРОН ТАРМАГЫНЫН НЕГИЗИНДЕ ТЕКСТТЕРДИН ТОНАЛДУУЛУГУНУН КЛАССИФИКАТОРУ

Н.М. Лыченко, А.В. Сороковая

Бул макалада тексттердин тоналдуулугунун классификациясы маселесин чечүү үчүн LSTM-тармагын колдонунун зарылдыгы негизделди жана ушул моделдин негизинде нейрон тармактык классификатору жана тексттердин тоналдуулугунун классификациялоо үчүн программалык система иштелип чыкты. Программалык система программалык модулдардын китепканасы болуп эсептелет жана ал тексттерди эсептөө алгоритмдери, сөздөрдү токенизациялоо, вектордук көрсөтмө, тексттердин тоналдуулугунун классификациясы жана алгоритмдердин жана WebAPI дын сапатын баалоо үчүн жооп берет. Ал иштелип чыккан китепкананы пайдалануу менен кардарлардын суроо-талабын тейлейт.

Түйүндүү сөздөр: тексттин тоналдуулугун аныктоо; классификациясынын милдети; токенизация; нейрон тармагы; LSTM-тармагы; программалык модулдардын китепканасы; WEB API.

CLASSIFIER OF TEXT SENTIMENT BASED ON LSTM-NEURAL NETWORK

N.M. Lychenko, A.V. Sorokovaya

The application of the LSTM network for solving the problem of text sentiment classification is justified. On the basis of this model a neural network classifier and a software system for classifying text tonality are developed. The software system consists from a library of software modules responsible for the algorithms of reading of text, tokenization, vector representation of words, text sentiment classification and quality assessment of algorithms, and the Web API that serves client requests using the developed library.

Keywords: sentiment analysis; classification; tokenization; neural networks; LSTM network; software module library; WEB API.

Введение. Автоматическая классификация текстов вызывает большой интерес в социологии, маркетинге, лингвистике, психологии и других сферах человеческой деятельности, поскольку позволяет разделять тексты по их смысловому содержанию с целью решения задач автоматического аннотирования, машинного перевода, составления каталогов текстов, классификации новостей и др. Особый интерес представляет автоматическая классификация

тональности текстов, заключающаяся в определении эмоциональной оценки авторами текстов информации, размещенной, например, в сети Интернет. Задача является актуальной, т. к. объемы информации в Сети стремительно возрастают, и прочесть и проанализировать мнения людей по каждой из исследуемых тем невозможно. Автоматический анализ тональности текстов позволяет сформировать обобщенное мнение о теме обсуждения без затрат времени со стороны человека. Дальнейшее исследование этого мнения является важной составляющей обеспечения эффективной политики и бизнеса, оценки эффективности маркетингового продвижения и рекламных кампаний, выбора целевой аудитории.

В задачах автоматической обработки естественного языка хорошо зарекомендовали себя искусственные нейронные сети – математические модели, используемые для обработки информации подобно нервной системе человека. Искусственная нейронная сеть (ИНС) [1, 2] – нелинейная вычислительная модель, используемая для решения задач машинного обучения. Любая архитектура ИНС состоит из искусственных нейронов – элементов обработки, имеющих структуру связанных друг с другом слоев. Каждый нейрон имеет взвешенные входы (синапсы), которые суммируются, и к сумме применяется функция активации (функция определения выхода нейрона при заданном входе). Могут применяться различные виды функции активации: линейная, ступенчатая, сигмоидная, тангенциальная, выпрямительная. Обучение нейронной сети происходит за счет настройки весов входов нейронов на основе минимизации ошибки, характеризующей разницу между полученными и требуемыми выходами сети. Чаще всего для обучения нейронных сетей используется метод обратного распространения ошибки [1] – одна из вариаций метода градиентного спуска. Достоинством нейросетевых методов является высокое качество классификации (при удачном подборе параметров). Однако они не лишены недостатков: возможная медленная сходимость или расходимость при использовании градиентных методов, необходимость более крупной обучающей выборки, низкая скорость обучения, сложная интерпретация результатов.

Приведем наиболее популярные архитектуры нейронных сетей, успешно используемые в задачах обработки естественного языка, в том числе классификации текстов:

- многослойный перцептрон (полносвязная сеть, в которой каждый нейрон слоя соединен с каждым нейроном следующего слоя), состоящий из входного, одного или более скрытых и одного выходного слоя [2];
- сверточная нейронная сеть, которая является вариантом многослойного перцептрона, и содержит так называемые сверточные слои, которые используют операцию свертки для входных данных и передают результат в следующий слой (эта операция позволяет сети быть глубже с меньшим количеством параметров [2]);
- рекурсивная нейронная сеть, или нейронная сеть древовидной структуры – тип глубокой нейронной сети, сформированный при применении модулей одного типа рекурсивно – выходной сигнал одного модуля подается на вход другому [3];
- рекуррентная нейронная сеть (вариант рекурсивной нейронной сети) [2], в которой связи между нейронами – направленные циклы, в результате чего слой сети получает информацию не только с предыдущего слоя сети, но и информацию о предыдущем состоянии самого себя.

Стоит учитывать, что сети прямого распространения позволяют анализировать только ограниченный контекст текста, в то время как рекуррентные – контекст произвольной длины, поскольку рекуррентные нейронные сети содержат обратные связи, а, значит, позволяют сохранять информацию о своих прошлых состояниях.

Однако обычные рекуррентные сети теряют способность связывать информацию по мере роста расстояния до контекста слова, поскольку могут учитывать только недавние прошлые состояния сети. Для решения этой проблемы создана специальная разновидность архитектуры рекуррентной нейросети – сеть долгой краткосрочной памяти (Long Short-Term Memory), LSTM-сеть [4]. LSTM-сеть позволяет обнаруживать как длинные, так и короткие шаблоны в данных, а также частично устраняет проблему исчезновения градиента.

Цель настоящей работы – применить модель LSTM-сети к решению задачи классификации тональности текста, поскольку именно

эта модель позволяет учитывать порядок слов в тексте и выявлять зависимости между далеко расположенными словами в контексте, и разработать на основе этой модели классификатор в виде программной системы.

Статья скомпонована следующим образом. В первой секции сформулирована задача классификации тональности текстов и этапы ее решения, во второй секции описан процесс построения и обучения классификатора на основе LSTM-сети, в третьей секции представлена архитектура нейросетевого классификатора и основные характеристики разработанной на его основе программной системы.

1. *Задача классификации тональности текстов и этапы ее решения.* Задача классификации в общем смысле – это разделение множества объектов на классы [5]. В данной работе задача классификации рассматривается как задача машинного обучения, при котором на основе обучающей выборки – конечного множества объектов, для которых классы заранее определены экспертом или иным способом, – автоматически определяется правило определения классовой принадлежности объектов вне обучающей выборки.

Математически задача классификации может быть сформулирована следующим образом:

- дано множество описаний объектов X и множество номеров (наименований) классов Y;
- описание объекта $x \in X$ представляет собой вектор признаков $x = (f_1(x), f_2(x),\ldots,f_n(x))$, называемый признаковым описанием объекта x;
- существует неизвестное отображение y^*: $X \to Y$, значения которого определены на обучающей выборке;
- требуется построить алгоритм a: $X \to Y$, способный классифицировать любой объект $x \in X$.

Классификация текстов, или классификация документов – одна из разновидностей задачи классификации, заключающаяся в отнесении текста к некоторым категориям на основании их содержания [6].

Основные этапы решения задачи классификации тональности текста на основе машинного обучения, следующие: предварительная обработка текстов, индексация текстов, построение и обучение классификатора, оценка качества классификации.

Предварительная обработка текста обязательно включает в себя токенизацию – выделение единиц языка в тексте (лексем, термов, токенов).

Также предварительная обработка текста может включать: приведение слов к одному регистру для исключения семантического различия между одинаковыми словами в разном регистре; удаление таких слов, как союзы, предлоги, артикли, т. е. семантически нейтральных слов (стоп-слов); удаление чисел или их замена на текстовый эквивалент; удаление пунктуации и излишних пробельных символов; нормализацию слов – выделение основы или корня слова (стемминг) или получение словарной формы слова (лемматизация), замена всех слов на нормализованную форму. Возможны и другие способы предобработки текста. Выбор сценария предобработки зависит от рода решаемой задачи и свойств выборки. В результате из текста выделяются все значимые слова, действительно определяющие его семантический смысл.

Далее необходимым шагом в задачах обработки текста на естественном языке является замена его некоторой числовой моделью в виде вектора его признаков – признакового описания текста, т. н. индексация текста [7]. Методы индексации активно развиваются, одни основываются на вероятностном распределении слов, другие на анализе контекста, однако все они обладают недостатками и сосредотачиваются лишь на некоторых характеристиках текстов, а, следовательно, могут по-разному работать на наборах текстов разного рода. В настоящей работе применены методы векторного представления слов для индексации текстов (LSA, Word2Vec, GloVe, ИНС), так как их основными преимуществами являются поддержка контекстуального сходства слов и отображение слов в низкоразмерное пространство. Однако описание этих методов, их программная реализация и сравнение эффективности в рамках данной статьи не представлены.

Для оценки качества классификации необходимо использовать специальные тестовые выборки. Тестовая выборка – размеченный набор данных, т. е. набор текстов с номером (наименованием) класса, к которому они относятся. Классифицируя объекты тестовой выборки, и сравнивая полученные классы с реальными классами,

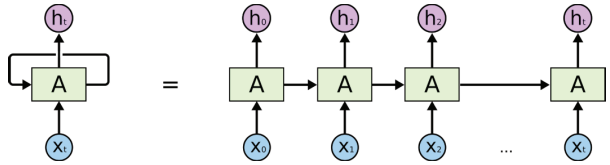

Рисунок 1 – Рекуррентная нейронная сеть в виде последовательности модулей

можно оценить качество классификации в виде численной метрики [8].

2. *Построение и обучение классификатора на основе LSTM-сети.* В настоящей работе для классификации использована модель LSTM-сети как вариант ИНС, позволяющий учитывать порядок слов в тексте и выявлять зависимости между далеко расположенными словами в контексте.

Рекуррентная нейронная сеть [4] может быть представлена в форме последовательности одинаковых модулей нейронной сети (рисунок 1).

В обычной рекуррентной сети модуль представляет собой один слой нейронов с обычной функцией активации. Модуль LSTM-сети представляет собой не один, а четыре слоя, которые взаимодействуют особым образом (рисунок 2). Прямоугольником здесь обозначен слой нейронной сети, кружком – поточечная операция, стрелками – поток передачи целого вектора, сходящимися стрелками – конкатенация векторов; условные обозначения при этом: x_t – входной вектор в момент времени t; h_t – выходной вектор в момент времени t; C_t – вектор состояний в момент времени t; W_k – матрица параметров слоя k, т. е. веса связей; b_k – вектор смещений выходов слоя k; f_t – вектор фильтра забывания в момент времени t; i_t – вектор входного фильтра в момент времени t; o_t – вектор выходного фильтра в момент времени t.

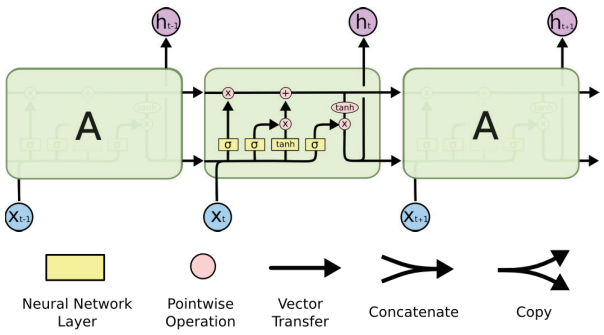

Рисунок 2 – Общая архитектура LSTM-сети

Ключевой компонент модуля LSTM-сети – вектор состояния. Вектор состояния регулируется специальными фильтрами, они управляют удалением (забыванием) и обновлением информации в нем. Сигмоидальный слой в фильтре определяет, какую долю информации перезаписать в векторе состояния (0 – не записывать ничего, 1 – записать все).

LSTM работает следующим образом.

1. Слой фильтра забывания определяет, какую информацию можно удалить из вектора состояния $C_{(t-1)}$ (рисунок 3, a). Для каждого числа из состояния возвращается число от 0 до 1 (где нулевое значение обозначает, что значение надо забыть):
$$f_t = \sigma\left(W_f \left[h_{t-1}, x_t\right] + bf\right).$$
2. Сигмоидальный слой входного фильтра определяет, какие значения следует обновить, а тангенс-слой строит вектор новых значений , которые могут быть добавлены в состояние (рисунок 3, b):
$$i_t = \sigma\left(W_i \left[h_{t-1}, x_t\right] + b_i\right),$$
$$\tilde{C}_t = \tanh\left(W_c \left[h_{t-1}, x_t\right] + b_c\right).$$
3. Для обновления состояния старое состояние необходимо умножить на f_t и прибавить к нему $i_t * C_t$ (рисунок 3, c).
4. Выход LSTM-сети представляет собой состояние C_t с примененной к нему функцией активации, при этом выходной фильтр определяет, какие именно элементы состояния выводить (рисунок 3, d):
$$h_t = o_t * \tanh\left(C_t\right),$$
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right).$$

Чтобы определить веса связей в нейросети, сеть необходимо обучить. В начале обучения веса связей инициализируются чаще всего случайным образом. Затем для каждой записи в обучающей выборке (содержащей текст и класс тональности: положительный/негативный) сеть определяет класс и сравнивает с тем классом, который нейросеть должна была получить. С помощью алгоритма обратного распространения ошибки, веса сети корректируются. Предварительно процессу обучения препроцессор преобразует каждый текст из выборки в последовательность векторов слов.

Для проверки обучения сети применяется перекрестная проверка на тестовой выборке.

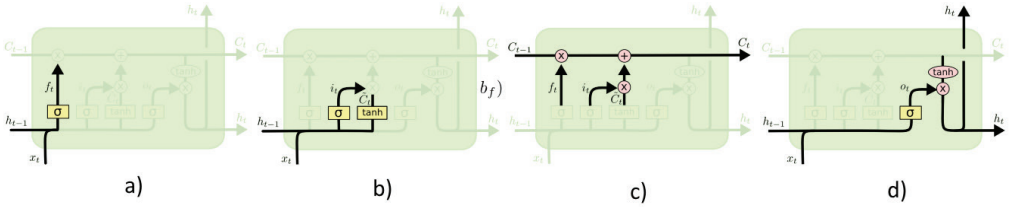

Рисунок 3 – Принцип работы LSTM-сети

Для этого исходная выборка (набор текстов) делится на n частей. Часть n-1 используются для обучения сети, а оставшиеся части используется для проверки сети. Этот процесс повторяется n раз так, что каждая из частей используется единожды в качестве тестовой. При этом вычисляются метрики ошибок, вычисленные на основе [8], которые усредняются и используются для оценки качества классификации.

3. *Реализация классификатора тональности текста на основе LSTM-сети.* В настоящей работе использована LSTM-сеть в связке со вспомогательными слоями. Архитектура сети представлена на рисунке 4 и представляет собой:

- входной слой, принимающий документ как последовательность индексов лексем – термов;
- слой векторного преобразования слов с фиксированными весами, где веса слоя преобразования задаются матрицей, *i*-ая строка которой представляет собой векторное представление *i*-го терма; служит для выбора вектора соответствующего терма; матрица строится на основе векторов, сгенерированных моделью векторного представления слов;
- слой регуляризации, который меняет некоторый процент значений выхода предыдущего слоя для предотвращения переобучения;
- сверточную сеть – последовательность сверточного и субдискретизирующего слоя с функцией максимума; используется для уменьшения количества входных параметров следующего слоя, повышает скорость обучения;
- слой LSTM как основной классификатор;
- выходной слой с сигмоидальной функцией активации.

Программная система для классификации тональности текстов содержит два основных компонента:

- библиотеку SentimentLib.dll, содержащую программные модули для считывания из файлов текстовых выборок, токенизации и векторного представления текстов, обучения и использования классификатора для анализа тональности текста, оценки качества алгоритмов (при разработке использованы научные библиотеки: NumPy, GenSim, Keras);
- Web API системы как отдельный исполняемый компонент, который обслуживает запросы клиента и использует библиотеку SentimentLib.dll классификации и сопутствующих процессов.

Система реализована в трехуровневой архитектуре и предоставляет интерфейсы: Web API для запросов клиента на анализ тональности, классификацию заданных текстов и выбор для этого классификатора и Web API – для запросов администратора системы на изменение конфигурации системы, управление обучением нейронных сетей, сохранение классификаторов, а также управление доступом других клиентов [9–11]. При отладке программной системы были использованы корпусы текстов на русском [12] и английском [13] языках.

Для функционирования программной системы необходимо, чтобы на серверной машине был установлен интерпретатор Python v. 3.5 и выше, на клиентской и серверных машинах был доступ к сети Internet, а также необходимо использование интерфейса Facebook Graph API для получения комментариев из сети Instagram [14].

Заключение. Таким образом, для решения задачи классификации тональности текстов выбрана модель LSTM-сети, поскольку именно эта модель позволяет учитывать порядок слов в тексте и выявлять зависимости между далеко расположенными словами в контексте. На основе этой модели разработан нейросетевой классификатор и программная система для классификации тональности текстов. Программная система представляет собой библиотеку программных модулей, отвечающих за алгоритмы

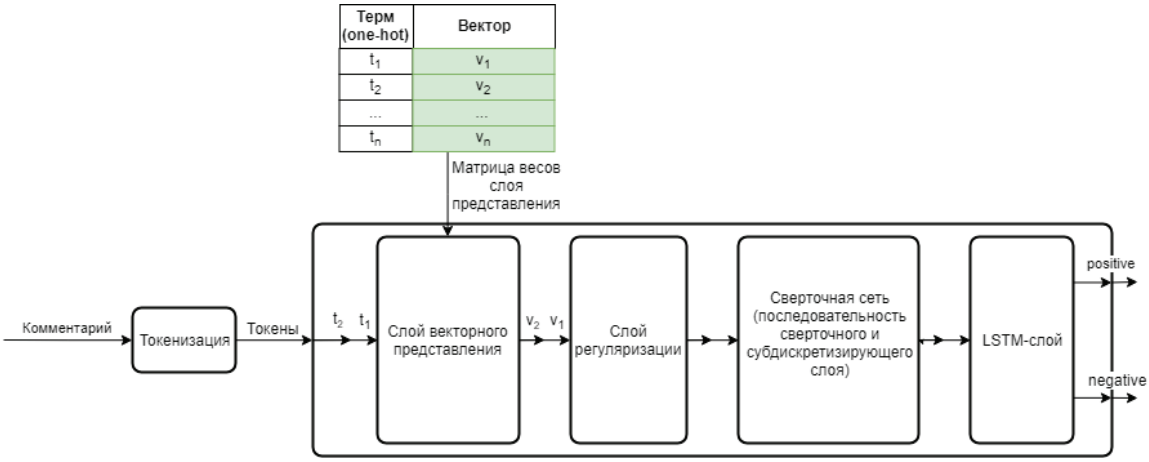

Рисунок 4 – Архитектура нейронной сети для анализа классификации текста

считывания текстов, токенизации, векторного представления слов, классификации тональности текстов и оценки качества алгоритмов, и Web API, который обслуживает запросы клиента и использует разработанную библиотеку классификации и сопутствующих процессов.

Разработанная программная система позволяет: создавать и обучать модели векторного представления слов русского и английского языков; обучать новый классификатор на основе созданных моделей векторного представления слов, использовать выбранный классификатор для определения тональности текстов из файлов и/или из комментариев под Instagram-постами; управлять конфигурацией системы (подключение новых методов векторного представления слов и классифицирующих алгоритмов); управлять доступом к системе (процедуры авторизации и аутентификации).

Литература

1. *Хайкин С.* Нейронные сети. Полный курс / С. Хайкин; пер. с англ. 2-е изд. М.: Изд. дом “Вильямс”, 2006. 1104 с.
2. 7 архитектур нейронных сетей для решения задач NLP Neurohive. URL: https://neurohive.io/ru/osnovy-data-science/7-arhitektur-nejronnyh-setej-nlp/ (дата обращения: 25.05.2018).
3. Deep Learning, NLP, and Representations Posted. URL: http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/ (дата обращения: 06.07.2018).
4. Understanding LSTM Networks. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (дата обращения: 14.09.2018).
5. *Барсегян А.А.* Методы и модели анализа данных: OLAP и Data Mining / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. СПб.: БХВ-Петербург, 2004. 336 с.
6. Классификация документов. Виды и методы. URL: https://studfiles.net/preview/4513655/page:6/ (дата обращения: 16.12.2017).
7. *Батура Т.В.* Методы автоматической классификации текстов / Т.В. Батура. Новосибирск: Новосиб. госуд. ун-т, 2017. Т. 30. № 1. С. 85–99.
8. Оценка классификатора (точность, полнота, F-мера). URL: http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html (дата обращения: 29.05.2018).
9. *Wilde E.* REST: From Research to Practice / E. Wilde, C. Pautasso // Springer Science & Business Media. 2011. 528 p.
10. *Erl Th., Carlyle B., Pautasso C., Balasubramanian R.* 5.1 // SOA with REST, 2017. URL: https://www.safaribooksonline.com/library/view/soa-with-rest/9780132869904/Tmpfs (дата обращения: 19.08.2018).
11. Протокол передачи гипертекста. URL: https://developer.mozilla.org/ru/docs/Web/HTTP (дата обращения: 13.01.2019).
12. Тренировочные корпусы текстов на русском языке. URL: http://study.mokoron.com/ (дата обращения: 09.02.2019).
13. IMDB Dataset of 50K Movie Reviews. URL: https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. (дата обращения: 12.02.2019).
14. Facebook Graph API. URL: https://developers.facebook.com/docs/graph-api/ (дата обращения: 03.04.2019).