УДК 519.712.2 DOI 10.18522/2311-3103-2023-5-138-151

Л.А. Гладков, Н.В. Гладкова, В.М. Курейчик

ПОДСИСТЕМА АВТОМАТИЧЕСКОГО АННОТИРОВАНИЯ ТЕКСТОВ НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ*

Рассматривается задача автоматического аннотирования текстов. Рассмотрена постановка задачи. Обоснована актуальность и важность разработки эффективных методов и программных систем для решения задачи автоматического реферирования текстов в современных информационных системах. Приведены определения понятий «данные» и знания». Описан перечень задач, относящихся в направлению Data Mining. Подробно описана задача Text Mining и существующие методы ее решения. Рассмотрена задача реферирования текстов. Выделены основные этапы решения задачи суммаризации. Описаны основные методы автоматической обработки текста, выделены их достоинства и недостатки. Подробно рассмотрены методы реферирования и квазиреферирования. Проведен сравнительный анализ эффективности различных методов реферирования и квазиреферирования, выделены их ключевые достоинства и недостатки. Приведено краткое описание архитектуры encoder-decoder с точки зрения использования данной архитектуры в разрабатываемом алгоритме автоматического реферирования текстов. Приведено описание модели рекуррентных нейронных сетей, отмечены достоинства и недостатки подобных моделей. Рассмотрены архитектуры рекуррентной нейронной сети применительно к решению задачи автоматического реферирования текстов. Приведено описание модифицированной модели рекуррентной нейронной сети – нейронной сети долгой краткосрочной памятью. Приведено описание предложенного алгоритма автоматического реферирования и значения настроек его основных параметров. Приведено описание разработанной программной подсистемы автоматического реферирования. Выполнено компьютерное моделирование и приведены результаты, полученные в ходе вычислительных экспериментов. Выполнена оценка качества полученных решений. Определены оптимальные параметры разработанной программной системы. Сформулированы направления продолжения исследований.

Реферирование текстов; суммаризация; методы реферирования и квазиреферирования; рекуррентные нейронные сети; токенизация; стемминг; нейронные сет долгой кратковременной памяти.

L.A. Gladkov, N.V. Gladkova, V.M. Kureichik

SUBSYSTEM FOR AUTOMATIC TEXT ANNOTATION BASED ON MACHINE LEARNING METHODS

This paper considers the problem of automatic text annotation. The formulation of the problem is considered. The relevance and importance of developing effective methods and software systems for solving the problem of automatic text summarization in modern information systems is substantiated. Definitions of the concepts "data" and knowledge are given." A list of tasks related to the Data Mining direction is described. The Text Mining problem and existing methods for solving it are described in detail. The problem of summarizing texts is considered. The main stages of solving the summation problem are highlighted. The main methods of automatic text processing are described, their advantages and disadvantages are highlighted. Abstractive summarization and extractive summarization methods are discussed in detail. A comparative analysis of the effectiveness of various abstracting and quasi-abstracting methods has been carried out, their key advantages and disadvantages have been highlighted. A brief description of the encoder-decoder architecture is given from the point of view of using this architecture in the developed algorithm

* Исследование выполнено за счет гранта Российского научного фонда № 22-21-00316, https://rscf.ru/project/22-21-00316/.

for automatic text summarization. A description of the model of recurrent neural networks is given, the advantages and disadvantages of such models are noted. The architecture of a recurrent neural network is considered in relation to solving the problem of automatic text summarization. A description of the modified model of a recurrent neural network – a neural network with long short-term memory – is given. A description of the proposed automatic abstracting algorithm and the settings of its main parameters are given. A description of the developed automatic abstracting software subsystem is given. Computer modeling is performed and the results obtained during computational experiments are presented. The quality of the solutions obtained was assessed. The optimal parameters of the developed software system are determined. Directions for continuing research are formulated.

Text summarization; text mining; abstractive summarization; extractive summarization methods; recurrent neural networks; tokenization; stemming; long short-term memory networks.

Введение. В современном мире информация играет ключевую роль во многих сферах. Количество информации растет в геометрической прогрессии, особенно в сети Internet. Мировая сеть содержит около 80-90% неструктурированной информации. Человечество старается оцифровать как можно большее количество такой информации в целях хранения, экономии места, удобства использования [1]. Для этого разрабатываются новые технологии хранения, создаются новые базы и хранилища данных. Сравнительно недавно появились и стали набирать популярность облачные хранилища ввиду удобства использования и хранения информации.

Далеко не вся подобная информация одинаково полезна и, более того, структурирована, что приводит к проблеме появления огромного количества данных, которые не поддаются ручной обработке. Для решения подобных проблем и получения профессиональных знаний используются методы data mining [2].

Стоит определить, что же такое «профессиональные знания». По сути – это сведения, которые характеризуют особенности деятельности и позволяют реализовать эту самую деятельность максимально эффективно. В связи с динамически развивающимися условиями рынка преимуществом современного специалиста является максимально широкий охват знаний в данной области. Например, для наиболее эффективного осуществления деятельности электронных магазинов необходимо обрабатывать большое количество неструктурированной информации (форумы, отзывы о продукции, информация о клиентах, пользовательский контент и т.д.). Поэтому актуальной является разработка предварительно обученной подсистемы, способной самостоятельно обрабатывающей пользовательские отзывы с целью получения короткого предложения, характеризующего эмоциональную окраску и смысл исходного экземпляра.

В зависимости от стоящих задач определяется подход к реферированию информации. В настоящее время существует большое количество алгоритмов для обработки неструктурированной информации. Одним из популярных подходов к решению данной задачи является использование нейронных сетей различных видов. Свою популярность они приобрели за счет эффективности работы в конкретной, узкоспециализированной сфере деятельности. Например, компания Google в 2018 представила новую технологию обработки естественного языка BERT (Bidirectional Encoder Representations from Transformers). Однако подобные системы имеют существенные недостатки, например, высокая требовательность к качеству информации, содержащейся в обучающей выборке.. Это подтверждает вывод о востребованности подобных систем на основе нейронных сетей новой архитектуры и, во-вторых, о необходимости их совершенствования.

Методы решения задачи автоматического реферирования. Data Mining (с англ. «добыча данных») – это автоматизированный поиск полезной информации среди огромного количества массивов неструктурированной информации. Основной задачей Data Mining является определение паттернов и взаимосвязей, выявле-

ния которых невозможно добиться при обычном анализе [2, 3]. Для решения подобных задач используют многочисленные математические и статистические алгоритмы. На данный момент существует множество алгоритмов (технологий), связанных с Data Mining – деревья решений, генетические алгоритмы, нейронные сети и производные.

Можно выделить три основных сферы применения Data Mining: Information Retrieval, Text Mining и Web Mining.

Information Retrieval («добыча данных») – группа методов, ориентированных на получение структурированных данных или же выборки данных меньшего размера [4–6]. Например, поисковая система, которая с помощью заданных алгоритмов позволяет получить необходимую информацию из большого массива документов. В качестве инструментов обычно выступают парсеры, методы индексации, фильтрации и сортировки данных.

Text Mining («добыча данных из текста») – это частный случай Data Mining, ориентированный на извлечение информации из текстов. В отличие от Information Retrieval, задачей в области Text Mining является анализ имеющихся данных с помощью математических методов, что позволяет получать новые знания [7]. Эта группа методов рассчитана на работу как частично обработанными, так и с «сырыми» данными.

Web Mining («добыча данных в Web») – набор методов и техник для получения данных из веб-источников [8]. Такие источники не являются текстовыми данными, соответственно и методы отличаются от Data Mining. Особенность в данном случае состоит в том, что в веб-ресурсах данные представлены в виде определенных форматов (Atom, SOAP, RSS, HTML), такие ресурсы располагают метаинформацией и информацией о структуре документа. Нужно также иметь ввиду наличие правил поисковой оптимизации (SEO).

Рассмотрим более подробно методы, относящиеся к группе Text Mining. Методы Text Mining направлены на превращение неструктурированных текстовых единиц в знания или же в подходящий для машинной обработки набор символов. Главными методами в Text Mining являются классификация и кластеризация, но далеко не всегда применяются именно они (зачастую они являются основой для различных надстроек и модификаций). Можно кратко выделить основные задачи, для решения которых используются методы Text Mining:

1. Извлечение понятий. Данная задача подразумевает повышение качества алгоритмов классификации, поиска и кластеризации с помощью получения новых «понятий» из текста. Например, необходимо создать базу данных из полезной информации, полученной в ходе анализа документов на естественном языке. Обычно в таких случаях осуществляется извлечение сущностей и связей, а также терминологии с последующим автореферированием [9].

2. Ответ на запросы (Question Answering). С помощью методов ответов на запросы мы можем дать возможность системе не только понять сам вопрос на естественном языке, но и предоставить пользователю понятный и вразумительный ответ. Примерный алгоритм состоит из поиска информации в частях документа, потенциально содержащих ответ, затем – фильтрация фраз, похожих на ответ, и, наконец, сам поиск правильного ответа.

3. Тематическое индексирование. Под «тематическим индексированием» изначально имелось ввиду определение для документов или запросов определенных индексов, которые могли бы отражать некие характерные особенности данных единиц. Но впоследствии это трансформировалось в задачу перевода текстовых единиц с естественного языка в формализованный вид, тогда как полученные описания представляют собой наборы ключевых слов и словосочетаний, отражающих тематическое содержание.

4. Поиск ключевых слов. В процессе поиска по ключевым словам ориентируются на результаты тематического индексирования для поиска документов, которые отвечают указанным требованиям (например, содержат пользовательские ключевые слова). Ключевыми словами в Text Mining можно назвать множество слов («терминов»), отражающих содержимое текста. Обычно метод используют для анализа частоты появления слов в тексте (основа для суммаризатора TF-IDF) [10].

5. Суммаризация (Text Summarization) [11]. С точки зрения человека реферирование и аннотирование текста – сложный вид интеллектуальной деятельности. Подобный вид деятельности требует большого количества временного ресурса. К тому же, в различных странах параллельно могут проходить исследования одного и того же материала, что связано с задержкой распространения информации в мире по разным причинам. Автоматизация подобных процессов – это выход из этой довольно сложной ситуации, который сэкономит время и ресурсы на обработку огромного количества текстов, многие из которых могут вообще не представлять никакой ценности для конкретного исследования.

Для эффективного извлечения скрытой информации из любого текста необходимо провести его предварительную обработку. В процессе такой обработки можно выделить три этапа [12]:

- ♦ очищение от знаков препинания – вариативный процесс, который зависит от поставленной задачи и используемого алгоритма;
- ♦ очищение от стоп-слов – комплексная задача. Состоит в удалении из текста лишних слов («шума»), которые негативно влияют на работу любого статистического метода. Как правило, это служебные части речи. В зависимости от используемого алгоритма может меняться и набор исключаемых частей речи;
- ♦ токенизация – самая сложная из всех этапов. Здесь, в первую очередь, необходимо разделить весь текст на составные части. Формат таких частей определяется алгоритмом. Как правило, текст разделяют на минимальные части – слова – и проводят дальнейшую обработку. В случае, когда необходимо получить список слов для каждого предложения текст разделяют на предложения с дальнейшим разложением на слова [13].
- ♦ стемминг – подразумевает приведение полученных слов к начальной форме. Более сложные системы включают алгоритмы, которые приводят к одной форме все синонимы.

Конечным результатом выполнения всех вышеперечисленных этапов является аннотация.

В зависимости от поставленной задачи методы автоматической обработки текста с целью получения какой-либо выдержки можно разделить на две группы – «extraction-based» и «abstraction-based» (извлечение информации и получение абстрактной информации). Более сложные структуры, к примеру - предложения, могут считаться ключевыми на основе совокупностей правил. Одна из причин считать предложение ключевым – наличие большого количества ключевых слов.

Для поиска ключевых единиц можно использовать несколько методов. Каждый из них по-своему эффективен в определенной ситуации и для определенных задач. Можно выделить три основных метода: статистический, позиционный, логико-семантический.

Статистический метод базируется на идее о том, что наиболее важные по смыслу слова встречаются в тексте чаще всего. Таким образом, предложение является ключевым, если в нем содержится определенное количество ключевых слов. Для частотного анализа используются различные статистические коэффициенты [14].

В позиционном методе ключевые единицы определяются в зависимости от расположения единицы в тексте. Как правило, это касается сложных лингвистических конструкций – предложения и абзацы. Например, предложение-заголовок вполне может являться ключевым элементом.

Последний метод – логико-семантический. Такой метод рассчитан на исследование структуры и семантики слов внутри текста. Задача состоит в выделении из текста предложения с наибольшим функциональным весом. Это зависит от множества различных факторов – наличия в предложении семантически значимых слов, взаимосвязей этого предложения с контекстом и т.д.

Итоговая генерация текста зависит от единиц, которые мы получим на выходе алгоритма: набор ключевых предложений или набор значимых слов (N-грамм)В любом случае главной задачей будет скрепление их в единую смысловую конструкцию. Смысловое разделение можно учесть непосредственно в ходе извлечения информации.

Реферирование и квазиреферирование. Можно выделить основные две задачи в процессе суммаризации: квазиреферирование (extractive summarization) и реферирование (abstractive summarization) [15].

1. *Квазиреферирование*. Цель квазиреферирования состоит в анализе входного текста (или выборки текстов) с последующим взвешиванием предложений или иных составных частей текста с целью определения наиболее значимых единиц. Получаемый в результате текст (summary) состоит из набора таких единиц. Методы квазиреферирования оперируют понятием «вес», который характеризует значимость лингвистической единицы для конкретного контекста [16]. Как правило, алгоритмы на основе такого подхода довольно просты для реализации и основаны на эвристиках. Основной недостаток данного метода состоит в возможной потере в результирующем тексте каких-либо семантических и синтаксических связей между выделенными единицами (например, предложениями).

Наиболее известным среди методов квазиреферирования является метод Latent Semantic Analysis (LSA). Он основан на построении матрицы с последующим ее сингулярным разложением. Поэтому каждое слово исходного текста подлежит взвешиванию. Способы определения веса предложения разнятся в зависимости от алгоритма. Получаемое в результате краткое содержание (аннотация) текста содержит наиболее значимые лингвистические единицы. Подобные графовые алгоритмы демонстрируют достаточно хорошую эффективность для решения конкретных задач.

К данной группе можно также отнести метод на основе определения веса TF-IDF [17]. TF-IDF – наиболее часто используемая мера определения значимости слова. Под мерой значимости мы понимаем вес единицы. TF-IDF – это произведение двух коэффициентов:

- TF (term frequency) – коэффициент, позволяющий измерить вес слова внутри одного текста (или частотность слова). Именно из-за этого показателя на этапе предварительной обработки текста необходимо избавиться от стоп-слов. Такие слова обычно обладают наибольшим показателем TF, но нулевой смысловой нагрузкой, что сказывается на эффективности взвешивания единиц с помощью этого коэффициента;
- IDF (inverted document frequency) – показатель, который позволяет оценить вес слова с учетом выборки текстов. Этот коэффициент хорошо себя проявляет при наличии задачи определения тематики текстов. Это инвертированный показатель частотности единицы внутри корпуса текстов. Количество совпадений в конкретном тексте роли не играет.

При перемножении данных коэффициентов можно получить общую характеристику слова (N-грамма). Полученный результирующий вес слова отражает смысловую важность слова внутри анализируемого текста с опорой на тексты данной тематики (или выборки) [18].

2. *Реферирование.* В основе данного подхода лежит определение основных смыслов текста и дальнейшая генерация краткого содержания на основе полученных знаний. В настоящее время в подобных системах используют различные виды нейронных сетей, для обучения которых используются наборы текстовой информации. Недостатком такого подхода является большой объем данных для обучения и необходимость их предварительной подготовки. Несмотря отмеченные недостатки, данный подход является наиболее перспективным из существующих. Как правило, для реализации метода реферирования используется архитектура encoder-decoder, а также transformer [19].

Методы квазиреферирования более просты в реализации по сравнению с методами реферирования. Они не нуждаются в обработанной и размеченной обучающей выборке. В большинстве случаев подобный подход предоставляет приемлемое качество полученного результирующего краткого содержания. Но для систем, работающих с большими массивами данных, требующих высокого качества аннотации и максимально исключающих участие человека, используются именно методы реферирования.

Рекуррентные нейронные сети. Рекуррентные нейронные сети (RNNs) – это модели, широко используемые для обработки текстовых единиц на естественном языке, а наиболее эффективной архитектурой является – архитектура encoder-decoder LSTM (Long Short-term Memory Networks). Задачи, решаемые с помощью таких моделей – это частотный анализ и генерация нового текста [20]. Подобные решения используются для решения группы проблем последовательностей – seq2seq. По сути, проблема seq2seq заключается в определении следующего значения в последовательности или же выводе метки входной последовательности. Чаще всего это обозначается в виде отношений типа «один к одному» или «многие к одному».

Также могут существовать и задачи более сложного типа, например, определения последовательности входных данных и прогнозирования последовательности данных на выходе.

Цель использования рекуррентных нейронных сетей состоит в получении последовательности при получении или применении информации. Традиционно считается, что все входы и выходы в нейронных сетях независимы друг от друга, но, к сожалению, такая структура подходит для решения далеко не всех задач. Логично, например, что для предсказания последующего слова в последовательности необходимо учитывать и предыдущие слова. Рекуррентные нейронные сети именно поэтому и называются рекуррентными, что они выполняют определенной действие к элементу последовательности, учитывая всю предыдущую последовательность. Подобную идею можно интерпретировать и по-другому – такие RNN системы имеют «память». Теоретически можно предположить, что длина входных последовательностей может быть неограниченной, но на практике все обстоит несколько иначе (рис. 1), где

x_t – вход для временного шага t;

s_t – представляет собой скрытое состояние для шага t и может интерпретироваться как «память». Это значение зависит от текущего входа и всех предыдущих состояний: x_t: $s_t = f(U x_t + W s_{t-1})$. В качестве функции f используется нелинейная *tanh* или же *ReLU*. s_{-1} используется для вычисления первого скрытого состояния и при инициализации получает значение нуля (нулевого вектора).

o_t – представляет собой выход для шага t. К примеру, для предсказания слова в последовательности можно представить выход в качестве выхода вероятностей в словаре $o_t = softmax(Vs_t)$.

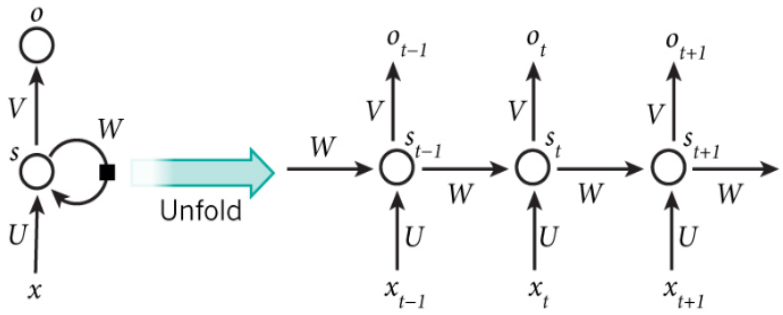

Рис. 1. Развертка рекуррентной нейронной сети

Таким образом, мы видим, что RNN можно развернуть в целую сеть. Допустим, что последовательность состоит из 10 слов. Тогда развертка будет включать 10 слоев.

Процесс обучения рекуррентной нейронной сети похож на обучение обычной нейронной сети. Обычно используется алгоритм обратного распространения ошибки (backpropagation), но с некоторыми оговорками. Исходя из того, что мы пользуемся одними и теми же параметрами на протяжении всех временных этапов сети, градиент зависит и от расчета текущего шага, и от предыдущих временных шагов. К примеру, если нам требуется вычислить градиент для $t = 3$, то мы «распространяем ошибку» на 2 шага, а затем суммируем полученные градиенты. Такая модификация backpropagation получила название «алгоритм обратного распространения ошибки сквозь время» или же «backpropagation through time» (BPTT). Недостаток такого обучения заключается в возможности анализировать только краткосрочные зависимости. Для обхода этих ограничений появились различные модификации, в том числе сети долгой краткосрочной памяти (Long Short-Term Memory, LSTM). От обычных RNN такие сети отличаются возможностью сохранять долгосрочные зависимости и способом вычисления скрытого состояния. Они рассчитаны на обучение долгосрочным зависимостям. Впервые LSTM описаны в работе Зеппа Хохрайтера и Юргена Шмихдхубера, опубликованной в 1997 году [21], после чего появилось множество модификаций данных сетей.

Если представить модуль обычной рекуррентной нейронной сети, то он будет выглядеть как один слой, содержащий функцию активации *tanh* (гиперболический тангенс). Структура LSTM выглядит похожим образом за исключением одной детали. Внутри блока содержатся четыре слоя с определенным взаимодействием (рис. 2).

Краеугольной идеей LSTM является наличие состояния ячейки (cell state). Информация попадает в блок, участвуя лишь в некоторых линейных преобразованиях. Так же LSTM может очищать состояние ячейки от информации, чему способствуют фильтры (gates).

В целом, LSTM – это огромный шаг в развитии нейронных сетей и рекуррентных нейронных сетей в частности. В настоящее время ведется работа над улучшением алгоритмов «attention», которые позволяют RNN брать данные из более крупного хранилища информации для более четкого определения контекста входящей последовательности.

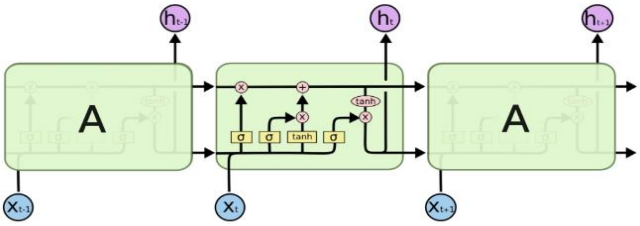

Рис. 2. Повторяющийся блок LSTM

Разработка алгоритма. Довольно часто отзывы клиентов могут быть громоздкими и описательными. Анализ таких отзывов вручную может отнимать значительное количество времени. В нашем случае целью является получение краткого оценочного реферата на основе входящего отзыва о продукте питания на английском языке. Для решения данной задачи была разработана система автоматического реферирования (суммаризатор), которая позволяет быстро определить основную идею отзыва и сформировать его краткое содержание. В качестве примера было решено использовать базу отзывов Amazon Fine Food с применением к нему модели seq2seq.

Использование модели «seq2seq» возможно для любой проблемы, которая связана с последовательной информацией. Нашей задачей является генерация реферата на основе отзывов пользователей. На вход будет поступать длинная последовательность слов, а на выходе мы должны получить краткое содержание. Соответственно, мы можем определить данную проблему seq2seq как «многие ко многим» [22].

При работе с нейронными сетями существует одна существенная проблема – невозможно сразу и заранее определить количество компонентов данной нейронной сети. В связи с этим были приняты примерные величины, используемые при обучении RNN. Например, было принято решение по общей структуре нейронной сети, а именно количество LSTM слоев равно 3. LSTM слои нам нужны для запоминания поступившей информации и взаимосвязи между ее элементами.

Для решения поставленных задач предложен следующий алгоритм обучения (рис. 3).

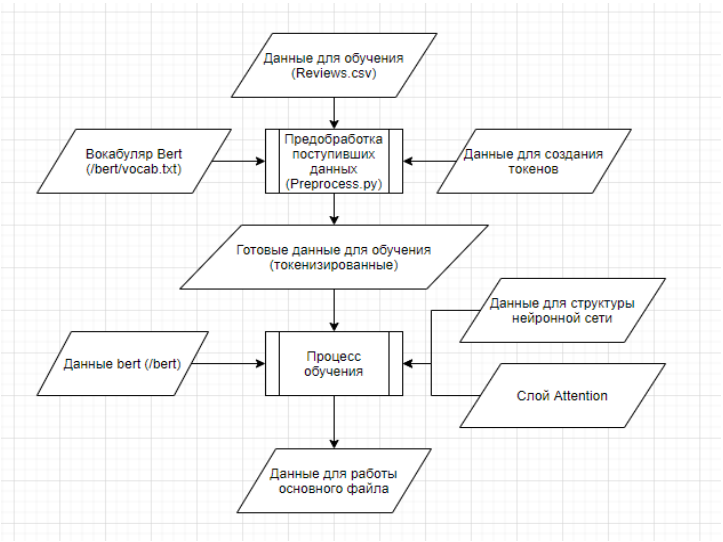

Рис. 3. Схема процесса обучения

Для выполнения предварительной обработки алгоритм использует следующие параметры:

- размеры обучающей и тестовой выборок - 0.9 и 0.9-1 соответственно;
- максимальная длина входной последовательности в токенах – 30.
- максимальная длина результирующей последовательности в токенах – 20.
- количество нейронов для слоев LSTM (влияет на качество и быстродействие) – 300.
- размерность эмбеддингов для декодера – 100.
- максимальное количество итераций, после которого обучение остановится, если до этого не сработает условие окончания – 50.

После прохождения каждого цикла обучения производится оценка по следующим показателям: Loss; Validation Loss.

На основе разницы этих показателей может быть принято решение об остановке обучения и сохранении наиболее удачного результата.

Описание разработанной программной системы. Для реализации системы был выбран язык программирования Python в среде разработки PyCharm Community Edition 2019. Данная среда разработки наиболее эффективно использует возможности языка программирования Python, который, в свою очередь, хорошо справляется с задачами Data Mining, особенно Text Mining. Разработка проводилась на операционной системе Microsoft Windows.

В связи с ресурсоемкостью процесса обучения также была использована среда Google Colab, которая предоставляет необходимые серверные GPU мощности.

Объектно-ориентированный язык Python имеет в своем распоряжении огромное количество библиотек для работы с текстовыми и статистическими данными.

Программная система располагает удобным установщиком библиотек Python, а также удобными инструментами разработки на этом языке.

Программная оболочка для пользователя представляет собой исполнительный файл для запуска приложения. Результатом запуска является отображение главной страницы приложения (рис. 4). Текст (до 30 символов) вводится в окне ввода сверху, результирующее предложение выводится ниже.

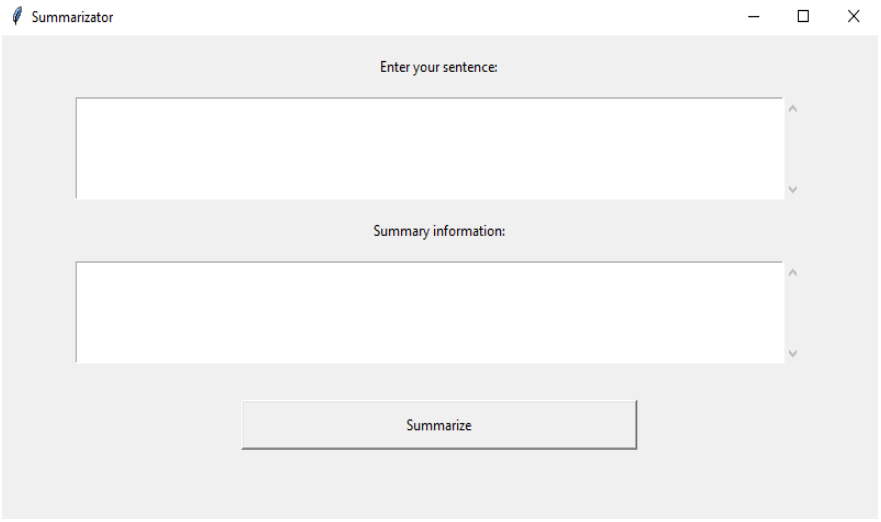

Рис. 4. Внешний вид оболочки

Результаты вычислительных экспериментов. Была проведена серия экспериментов для тестирования обучаемой модели. Изменялись различные параметры: количество последовательностей в выборке, количество нейронов для LSTM слоев, количество последовательностей, обрабатываемых единовременно и т.д. Количество единиц в выборке варьировалось от 10000 до 100000. Количество нейронов для LSTM слоев – от 50 до 200. Количество единовременно обрабатываемых последовательностей изменялось от 50 до 300.

В результате экспериментов выяснилось, что некоторые параметры невозможно повысить ввиду слабости машины для обработки. Было принято решение перенести обучение сети в Google Colab, где есть возможность обучать нейронную сеть на мощностях Google. Но для максимального качества работы алгоритма и этого оказалось недостаточно. На данный момент система обучена со следующими параметрами:

- размер выборки – 100000 отзывов;
- количество единовременно обрабатываемых отзывов – 200;
- количество нейронов для LSTM – 200;
- максимальное количество эпох – 50.

При данных параметрах был достигнут очень хороший результат, но стоит учесть, что обучение не прервалось автоматически ввиду большой разницы ошибок loss и val_loss. Результаты процесса обучения нейронной сети в Google Colab в виде диаграммы (рис. 5).

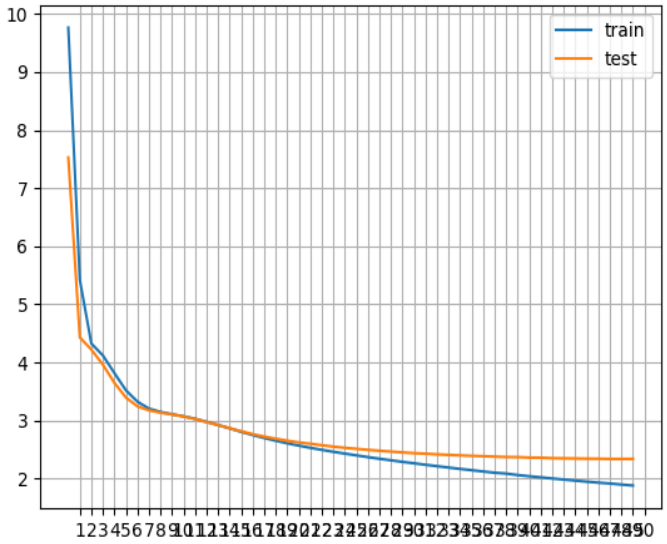

Рис. 5. Результаты обучения

На диаграмме можно увидеть зависимость loss и val_loss. При сильной разнице графиков происходит переобучение. Такого в нашем примере не наблюдается.

Оценить практическое качество работы алгоритма в данном конкретном случае довольно сложно. К примеру, очень популярная метрика ROUGE очень чувствительна к количеству слов, имеющих совпадение с ручным рефератом. Поэтому было принято решение использовать human evaluation.

Чтобы оценить результаты на практике было отобрано 10 предложений (отзывов) по адресу https://www.foodnetwork.com/recipes/anne-thornton/fabulous-fudge-recipe2-1924263.

Для определения качества была использована шкала цифр от 1 до 5. Оценивание происходило по двум параметрам: «fluency» и «adequacy». Под «adequacy» мы понимаем то, насколько близко по смыслу удалось машинному переводу приблизиться к экспертному. Шкала выглядит следующим образом:

1. Нет смысла.
2. Мало смысла.
3. Неполный смысл.
4. Достаточно смысла.
5. Смысл передан полностью.

Параметр «fluency» же отражает то, насколько правильно сформулировано предложение с точки зрения грамматики, орфографии, выбора слов и стилистики. Для этого показателя неважен смысл и первоисточник. Далее перечислены критерии для «fluency»:

1. Непонятно.
2. Звучит странно.
3. Речь не носителя.
4. Хорошо.
5. Отлично.

В столбцах «Adequacy» и «Fluency» указаны оценки всех десяти экспертов. Также имеются столбцы со средними показателями adequacy и fluency, на основании которых и можно оценить качество работы системы.

Как можно заметить, система очень хорошо справляется с конструированием последовательностей в стилистическом плане (Fluency), а некоторые неточности в семантическом плане могут быть исправлены различными путями: увеличение обучающей выборки с дальнейшей корректировкой параметров, изменение метода генерации последовательности (на данный момент применяется greedy search, возможно заменить на beam search.

Было принято решение увеличить количество предложений для теста до 40. Наглядно эффективность работы системы можно увидеть в виде диаграммы (рис. 6):

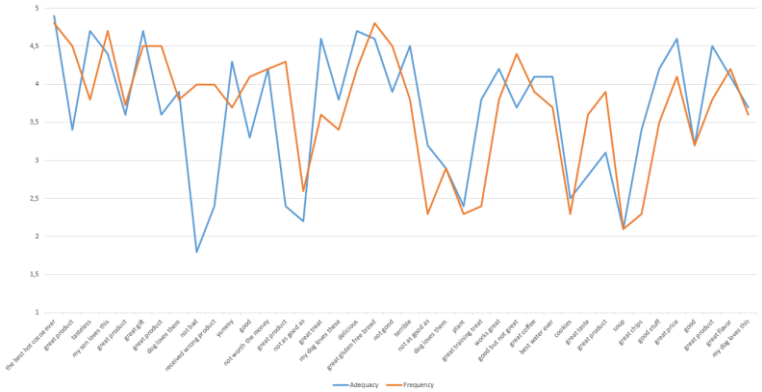

Рис. 6. График эффективности

На диаграмме представлены графики среднего значения adequacy и fluency для каждого результата из 40. Также, можно заметить, что оценка параметра adequacy (смысл предложения) имеет практически стабильную дисперсию (значение дисперсии равно 0,714), а у fluency - 0,578. Исходя из стоящих задач, можно сделать вывод, что параметр «adequacy» имеет большее значение, так как отзывы сами по себе не обладают грамматической или стилистической грамотностью.

Заключение. Задача разработки новых эффективных методов автоматического реферирования текстов становится все более востребованной по мере роста объемов обрабатываемой информации и необходимости скорости ее обработки. В зависимости от конкретной задачи применяются как алгоритмы для извлечения информации, так и для генерации совершенно новой. Все более активно для решения такого рода задач используются нейронные сети, но они требуют большого объема данных для обучения и точности калибровки. Так, например, компания Google в октябре 2019 года реализовала новый метод обработки естественного языка BERT (Bidirectional Encoder Representations from Transformers) на основе нейросетей новой архитектуры («трансформеры»).

В данной статье предложен подход к решению задачи автоматического реферирования отзывов о продуктах питания. Данный подход основан на использовании рекуррентной нейронной сети с определенной архитектурой и внедренным слоем BERT. Была разработана программная оболочка для демонстрации работы алгоритма.

Был проведен ряд экспериментов с параметрами сети и объемом обучающей выборки, что, в конечном итоге, позволило установить наиболее оптимальные настройки для эффективной работы алгоритма.

Определены возможности улучшения результатов работы системы за счет увеличения обучающей выборки, а также изменения метода поиска при построении результирующей последовательности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Мордвинов А.В.* Разработка и исследование модели текста для его категоризации: автореф. дисс. … канд. техн. наук: 05.13.01. – Н. Новгород, 2010. – 25 с.
2. *Тревгода С.А.* Методы и алгоритмы автоматического реферирования текста на основе анализа функциональных отношений: автореф. дисс. … канд. техн. наук: 05.13.01. – СПб., 2009. – 19 с.
3. *Лукашевич Н.В.* Модели и методы автоматической обработки неструктурированной информации на основе базы знаний онтологического типа: автореф. дисс. … канд. техн. наук: 05.25.05. – М., 2014. – 32 с.
4. *Van Lierde H., Chow T.W.S.* Query-oriented text summarization based on hypergraph transversals // Information Processing and Management. – 2019. – Vol. 56, No. 4. – P. 1317-1338.
5. *Greengrass E.* Information Retrieval: A Survey: – University of Maryland. 2000. – 225 p.
6. *Manning D., Raghavan C., Schütze H.* Introduction to Information Retrieval: Cambridge. – England. 2008.
7. *Alguliev R.M., Isazade N.R., Abdi A., Idris N.* COSUM: Text summarization based on clustering and optimization // Expert Systems. – 2019. – Vol. 36, No. 1.
8. *Харламов А.* Технология автоматического смыслового анализа текстов TextAnalyst // Вестник Московского государственного лингвистического университета. – 2014. – С. 234-244.
9. *Хоай Л., Тузовский А.Ф.* Семантическое аннотирование документов в электронных библиотеках // Известия Томского политехнического университета. – 2013. – С. 157-164.
10. *Харламов А.* Когнитивный подход к смысловому анализу текстов // Вестник Московского государственного лингвистического университета. – 2013. – Т. 13, № 673. – С. 196-205.
11. *Gupta V.. Bansal N., Sharma A.* Text summarization for big data: A comprehensive survey // Lecture Notes in Networks and Systems. – Delhi, 2019. – Vol. 56. – P. 503-516.
12. *Anam S.A., Muntasir Rahman A.M., Sleheen N.N., Arif H.* Automatic text summarization using fuzzy C-Means clustering // 2018 Joint 7th International Conference on Informatics, Electronics and Vision and 2nd International Conference on Imaging, Vision and Pattern Recognition. – Kitakyushu, 2018. – P. 180-184.
13. *Chua S., Kulathuramaiyer N., Ranaivo-Malancon B., Iboi H.* A comparative Study of Sentiment-Based Graphs of Text Summaries // 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences. – Sarawak, 2018.

14. *Siddiqui T.* Generating abstractive summaries using sequence to sequence attention model // 2018 International Conference on Frontiers of Information Technology. Proceedings. – Karachi, 2018. – P. 212-217.
15. *Sonawane S., Ghotkar A., Hinge S.* Context-based multi-document summarization // Advances in Intelligent Systems and Computing. – 2018. – Vol. 812. – P. 153-165.
16. *Alwis V.* Intelligent E-news summarization // 18th International Conference on Advances in ICT for Emerging Regions. – Colombo, 2018. – P. 189-195.
17. *Joshi A., Mehta K., Gupta N., Valloli V.K.* Data generation using sequence-to-sequence // 2018 IEEE Recent Advances in Intelligent Computational Systems. – Pune, 2018. – P. 108-112.
18. *Gigioli P., Sagar N., Rao A., Voyles J.* Domain-Aware Abstractive Text Summarization for Medical Documents // Proceedings 2018 IEEE International Conference on Bioinformatics and Biomedicine. – New York. 2018. – P. 2338-2343.
19. *Mahajani A., Pandya V., Maria I., Sharma D.* Ranking-Based Sentence Retrieval for Text Summarization // 2018 2nd International Conference on Smart Innovations in Communications and Computational Sciences. – Mumbai, 2018. – P. 465-474.
20. *Kirmani M., Manzoor Hakak N., Mohd M., Mohd M.* Hybrid text summarization // 2nd International conference of the series Soft Computing: Theories and Applications. – 2017. – Kuruhshetra, 2017. – P. 63-73.
21. *Hochreiter S.; Schmidhuber J.* Long short-term memory // Neural Computation: journal. – 1997. – Vol. 9, No. 8. – P. 1735-1780. – DOI: 10.1162/neco.1997.9.8.1735. – PMID 9377276.
22. *Гладков Л.А., Гладкова Н.В., Бова В.В.* Метод автоматического аннотирования текстов на основе гибридных интеллектуальных технологий // Информатизация и связь. – 2022. – № 2. – С. 54-60.

REFERENCES

1. *Mordvinov A.V.* Razrabotka i issledovanie modeli teksta dlya ego kategorizatsii: avtoref. dis. … kand. tekhn. nauk [Development and research of a text model for its categorization: abstract of cand. of eng. sc. diss.]: 05.13.01. Nizhniy Novgorod, 2010, 25 p.
2. *Trevgoda S.A.* Metody i algoritmy avtomaticheskogo referirovaniya teksta na osnove analiza funktsional'nykh otnosheniy: avtoref. dis. … kand. tekhn. nauk [Methods and algorithms for automatic text summarization based on the analysis of functional relationships: abstract of cand. of eng. sc. diss.]: 05.13.01. St. Petersburg, 2009, 19 p.
3. *Lukashevich N.V.* Modeli i metody avtomaticheskoy obrabotki nestrukturirovannoy informatsii na osnove bazy znaniy ontologicheskogo tipa: avtoref. diss. … kand. tekhn. nauk [Models and methods for automatic processing of unstructured information based on an ontological knowledge base: abstract of cand. of eng. sc. diss.]: 05.25.05. Moscow, 2014, 32 p.
4. *Van Lierde H., Chow T.W.S.* Query-oriented text summarization based on hypergraph transversals, *Information Processing and Management*, 2019, Vol. 56, No. 4, pp. 1317-1338.
5. *Greengrass E.* Information Retrieval: A Survey: University of Maryland. 2000, 225 p.
6. *Manning D., Raghavan C., Schütze H.* Introduction to Information Retrieval: Cambridge. England. 2008.
7. *Alguliev R.M., Isazade N.R., Abdi A., Idris N.* COSUM: Text summarization based on clustering and optimization, *Expert Systems*, 2019, Vol. 36, No. 1.
8. *Kharlamov A.* Tekhnologiya avtomaticheskogo smyslovogo analiza tekstov TextAnalyst [Technology for automatic semantic analysis of texts TextAnalyst], *Vestnik Moskovskogo gosudarstvennogo lingvisticheskogo universiteta* [Bulletin of the Moscow State Linguistic University], 2014, pp. 234-244.
9. *Khoay L., Tuzovskiy A.F.* Semanticheskoe annotirovanie dokumentov v elektronnykh bibliotekakh [Semantic annotation of documents in electronic libraries], *Izvestiya Tomskogo politekhnicheskogo universiteta* [News of Tomsk Polytechnic University], 2013, pp. 157-164.
10. *Kharlamov A.* Kognitivnyy podkhod k smyslovomu analizu tekstov [Cognitive approach to semantic analysis of texts], *Vestnik Moskovskogo gosudarstvennogo lingvisticheskogo universiteta* [Bulletin of the Moscow State Linguistic University], 2013, Vol. 13, No. 673, pp. 196-205.
11. *Gupta V.. Bansal N., Sharma A.* Text summarization for big data: A comprehensive survey, *Lecture Notes in Networks and Systems*. Delhi, 2019, Vol. 56, pp. 503-516.

12. *Anam S.A., Muntasir Rahman A.M., Sleheen N.N., Arif H.* Automatic text summarization using fuzzy C-Means clustering, *2018 Joint 7th International Conference on Informatics, Electronics and Vision and 2nd International Conference on Imaging, Vision and Pattern Recognition*. Kitakyushu, 2018, pp. 180-184.
13. *Chua S., Kulathuramaiyer N., Ranaivo-Malancon B., Iboi H.* A comparative Study of Sentiment-Based Graphs of Text Summaries, *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences*. Sarawak, 2018.
14. *Siddiqui T.* Generating abstractive summaries using sequence to sequence attention model, *2018 International Conference on Frontiers of Information Technology. Proceedings*. Karachi, 2018, pp. 212-217.
15. *Sonawane S., Ghotkar A., Hinge S.* Context-based multi-document summarization, *Advances in Intelligent Systems and Computing*, 2018, Vol. 812, pp. 153-165.
16. *Alwis V.* Intelligent E-news summarization, *18th International Conference on Advances in ICT for Emerging Regions*. Colombo, 2018, pp. 189-195.
17. *Joshi A., Mehta K., Gupta N., Valloli V.K.* Data generation using sequence-to-sequence, *2018 IEEE Recent Advances in Intelligent Computational Systems*. Pune, 2018, pp. 108-112.
18. *Gigioli P., Sagar N., Rao A., Voyles J.* Domain-Aware Abstractive Text Summarization for Medical Documents, *Proceedings 2018 IEEE International Conference on Bioinformatics and Biomedicine*. New York. 2018, pp. 2338-2343.
19. *Mahajani A., Pandya V., Maria I., Sharma D.* Ranking-Based Sentence Retrieval for Text Summarization, *2018 2nd International Conference on Smart Innovations in Communications and Computational Sciences*. Mumbai, 2018, pp. 465-474.
20. *Kirmani M., Manzoor Hakak N., Mohd M., Mohd M.* Hybrid text summarization, *2nd International conference of the series Soft Computing: Theories and Applications*, 2017. Kuruhshetra, 2017, pp. 63-73.
21. *Hochreiter S.; Schmidhuber J.* Long short-term memory, *Neural Computation: journal*, 1997, Vol. 9, No. 8, pp. 1735-1780. DOI: 10.1162/neco.1997.9.8.1735. PMID 9377276.
22. *Gladkov L.A., Gladkova N.V., Bova V.V.* Metod avtomaticheskogo annotirovaniya tekstov na osnove gibridnykh intellektual'nykh tekhnologiy [Method for automatic annotation of texts based on hybrid intelligent technologies], *Informatizatsiya i svyaz'* [Informatization and communication], 2022, No. 2, pp. 54-60.

Статью рекомендовала к опубликованию д.т.н., профессор Л.С. Лисицина.

Гладков Леонид Анатольевич – Южный федеральный университет; e-mail: lagladkov@sfedu.ru; г. Таганрог, Россия; кафедра САПР; профессор.

Гладкова Надежда Викторовна – e-mail: nvgladkova@sfedu.ru; кафедра САПР; старший преподаватель.

Курейчик Виктор Михайлович – e-mail: vmkureychik@sfedu.ru; кафедра САПР; профессор.

Gladkov Leonid Anatol'evich – Southern Federal University; e-mail: lagladkov@sfedu.ru; Taganrog, Russia; the department of CAD; professor.

Gladkova Nadezhda Viktorovna – e-mail: nvgladkova@sfedu.ru; the department of CAD; senior teacher.

Kureichik Viktor Mikhaylovich – e-mail: vmkureychik@sfedu.ru; the department of CAD; professor.