УДК 004.032.26

НЕЙРОННАЯ СЕТЬ ДЛЯ ГЕНЕРАЦИИ ВОПРОСОВ К ТЕКСТУ

В. В. Астрашаб
Студент БГУИР

М. А. Калугина
Доцент кафедры информатики, кандидат физико-математических наук, доцент

Д. А. Клебанов
Студент БГУИР

Д. С. Совпель
Студент БГУИР

К. И. Акулич
Студент БГУИР

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: u.astraszab@gmail.com, dmitriy.klebanov@gmail.com, sovspace@gmail.com, kirillakulch0 @gmail.com, marina_kalugina@list.ru

В. В. Астрашаб

В 2018 году окончил Могилевский государственный областной лицей №1. Принимал активное участие в олимпиадном движении, становился победителем олимпиад по математике, программированию, английскому языку и астрономии.

Д. А. Клебанов

В 2018 году окончил гимназию №2 города Могилева. В школьное время неоднократно становился победителем олимпиад по математике, программированию и физике. Победитель множества студенческих олимпиад по программированию, в том числе в четвертьфинале международной студенческой олимпиады (ACM ICPC).

Д. С. Совпель

В 2018 году окончил Кобринскую среднюю школу №7, где принимал активное участие в олимпиадном движении по математике и географии. В университете продолжает участвовать в олимпиадах: среди достижений – полуфинал BSUIR Open и диплом 2-й степени на олимпиаде БГУИР по математике.

К. И. Акулич

В 2018 году окончил СШ №2 г. Кировска. Принимал участие в олимпиадном движении, становился победителем олимпиад по математике и физике.

М. А. Калугина

После окончания мехмата БГУ, затем аспирантуры при кафедре САПР ФПМ БГУ работала в ИТК АН БССР. В настоящее время — доцент кафедры информатики БГУИР.

Аннотация. В данной работе рассмотрен алгоритм автоматической генерации вопросов к тексту, основанный на применении нейронных сетей. Описаны основные подходы к решению проблемы, устройство модели и реализация, приведены результаты работы алгоритма и перспективы для улучшения выбранного подхода.

Ключевые слова: генерация вопросов, большие данные, машинное обучение, нейросети, Natural Language Processing.

Введение

С развитием вычислительной техники и с появлением возможности автоматизации рутинных процессов человеческая деятельность во многих областях стала упрощаться. Большой скачок совершил учебный процесс в сфере образования: он не только перешел в Интернет, но и, в целом, стал более удобным как для преподавателей, так и для обучающихся.

Тесты и вопросы - одна из форм закрепления и контроля знаний. Чем более точно они подобраны, тем эффективнее можно реализовать процесс обучения. Автоматизированная система по составлению вопросов могла бы улучшить их качество и увеличить количество, тем самым покрывая больше деталей и тем программы обучения. Студенты при помощи системы могли бы сами генерировать вопросы по изученному материалу в качестве самопроверки.

Более того, эффективная генерация вопросов - открытая научная проблема, решение которой может помочь лучше понять работу мозга и механизмы языковых способностей человека. Генерация вопросов состоит из двух очень сложных процессов: анализа лексического и синтаксического контекста исходного предложения и построения по всем правилам языка нового предложения, которое основано на уже обработанной информации.

Одним из подходов к решению такой задачи является синтаксический анализ всех слов в предложении. На основе этой информации выделяются существительные или другие именные части речи, к которым и будет ставиться вопрос. Такой подход можно считать неэффективным из-за того, что не учитывается синтаксический и лексический контексты в исходном предложении. Результат такого алгоритма - вопросы к подлежащему, что также говорит о его недостаточной эффективности [1, 2].

Задачи по обработке синтаксического и лексического контекста, а также формированию предложений могут решаться при помощи методов машинного обучения, а именно: рекуррентных нейронных сетей для обработки естественного языка (natural language processing). Если провести анализ постановки задачи в контексте машинного обучения, то ее можно сформулировать как обучение с учителем (supervised learning). Подобные задачи генерации нового текста по заданному решаются с помощью использования энкодеров и декодеров. Особенностью обучения нейронной сети является потребность в больших объемах данных для обучения и в больших вычислительных мощностях для их обработки.

Описание модели

Модель состоит из двух частей.

– Рекуррентная нейронная сеть (RNN), которая обнаруживает слова, которые с наибольшей вероятностью являются ответами на потенциальные вопросы.

– Кодирующая-декодирующая нейронная сеть (encoder-decoder), которая генерирует вопросы к словам, выбранным RNN как наиболее вероятные ответы [4].

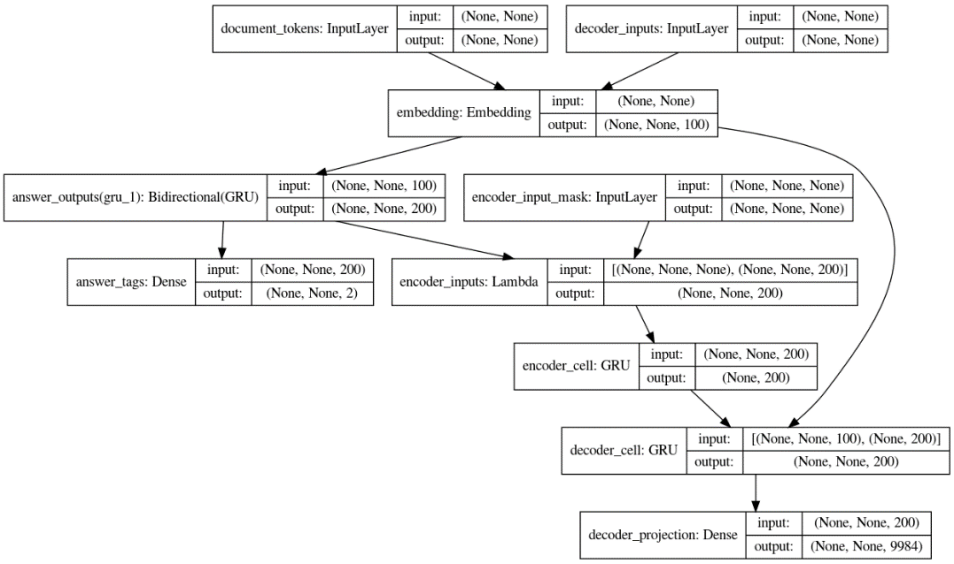

Схема *1.* – Архитектура модели

Для примера рассмотрим текст из набора новостей CNN:

“They accepted the Liverpool captain 's version that he acted in self defense in punching businessman Marcus McGhee . The 29 year old was the only one of the seven defendants in the case to be cleared after an incident which was described by judge Henry Globe as an explosion of violence.”

После обработки данного текста первая модель должна выбрать наиболее вероятные слова, такие как Marcus McGhee, 29 year old, seven defendants, judge Henry Globe.

Затем вторая модель по выбранным словам генерирует вопрос. Например, по сочетанию слов “29 year old” сеть должна генерировать вопрос: “How many years old was the businessman?“

Векторизированный текст подается рекуррентной нейросети, которая обучается предсказывать, является ли это слово ответом на некоторый вопрос. Рекуррентная нейронная сеть состоит из двунаправленной цепи (рисунок 2) GRU-ячеек. Эти ячейки (рисунок 1) последовательно принимают каждое слово из текста и результат обработки текста соседними ячейками. Это позволяет учитывать связь между близкими словами в тексте. Двунаправленность позволяет учитывать не только предыдущие слова текста, но и последующие при обработке каждого слова.

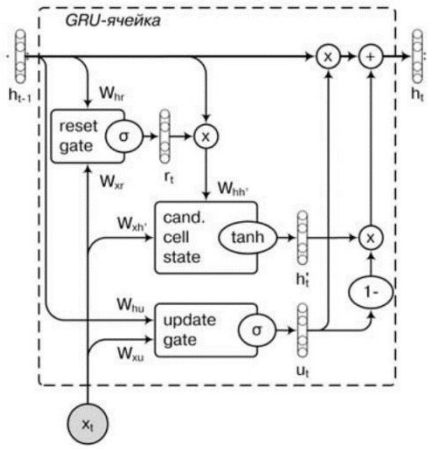

Рисунок *1.* – Внутреннее строение GRU-ячейки [3]

Веса этой нейросети обучаются с помощью алгоритма обратного распространения ошибки во времени (backpropagation through time). Ошибка измеряется на тренировочном наборе данных, где слова, помеченные ответами хотя бы в одном примере, помечаются 1, а остальные - 0. Оптимизируемая функция ошибки - бинарная кросс-энтропия.

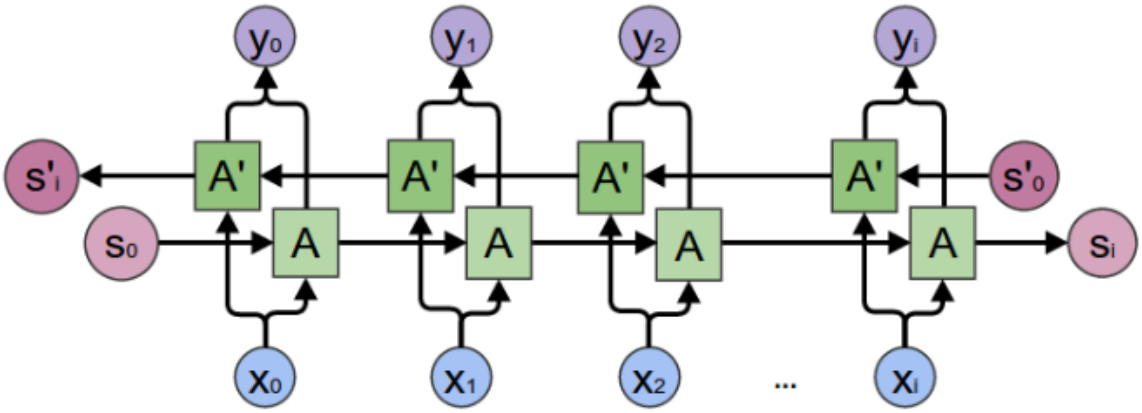

Рисунок *2*. – Двунаправленная сеть GRU-блоков, где x_i - векторы слов текста, y_i - предсказанные моделью вероятности того, что слово - это ответ на некий вопрос

Кодирующая-декодирующая нейронная сеть получает на вход предложение и маску с ответами, отделяющую слова, к которым нужно сгенерировать вопрос. Маска может быть составлена пользователем или автоматически на основе предсказаний предыдущего слова. Кодирующий слой (encoder) кодирует последовательность векторов слов ответа на потенциальный вопрос в один вектор, который далее будет “раскодирован” следующим слоем в вопрос.

Затем следующий слой (decoder) по вектору закодированного ответа со слоя encoder и исходному тексту последовательно генерирует слова вопроса. Генерация всегда начинается с токена <START>, после чего модель возвращает для слов из словаря вероятность того, что данное слово будет следующим в вопросе. Из полученных вероятностей выбирается максимальная, и это слово становится следующим словом вопроса. Процесс происходит до тех пор, пока модель не решит, что наиболее вероятным следующим словом будет токен <END>.

Слои encoder и decoder также состоят из GRU-ячеек и обучаются на ошибке отклонения сгенерированного вопроса от вопроса из тренировочного набора данных посредством алгоритма обратного распространения ошибки.

Реализация и обучение модели

Нейронные сети реализованы на языке Python с использованием библиотеки Keras. Для обучения модели был использован набор данных с новостями CNN и вопросами для них, состоящий из более чем 100000 примеров. Данный набор находится в открытом доступе в интернете и заранее разделён авторами на наборы для обучения (92000 примеров), валидации модели (5000) и финального тестирования (5000). Каждый пример из набора данных содержит номер новости, исходный текст, подходящий вопрос и номера слов в тексте, которые являются ответом на этот вопрос.

При подготовке данных для обучения наиболее часто встречающиеся слова заменены на векторы GloVe (Global Vectors for Word Representation), которые позволяют представить слова в виде, пригодном для обработки нейросетью, сохраняя лексическую взаимосвязь между словами. Затем такие слова пронумерованы, а реже встречающиеся слова заменены и помечены как <UNK> (неизвестное слово). На схеме 1 за этот процесс отвечает блок “Embedding”.

Для обучения весов модели использовалась модификация алгоритма стохастического градиентного спуска - алгоритм Adam. Оптимизируемая функция ошибки для нейронной сети не является выпуклой, поэтому использовались алгоритмы, которые находят локальный минимум. Алгоритм Adam быстрее и обеспечивает лучшую сходимость, чем другие модификации градиентного спуска. Также использовалось обучение по мини-батчам. Данные случайным образом разделялись на группы по 128 примеров (такая группа называется мини-батч от англ. mini-batch), и для каждой группы совершалась итерация алгоритма Adam. Период обучения, в течение которого модель будет обучена на всех данных из тренировочного набора по одному разу, называется эпохой. Наша модель обучалась в течение 4 эпох, то есть каждый пример из тренировочного набора был использован 4 раза. В течение этих 4 эпох можно наблюдать за ошибкой модели на текущем мини-батче тренировочных и валидационных данных. При успешном обучении ошибка должна снижаться одновременно для тренировочных и валидационных данных. Если ошибка снижается только на тренировочных данных, но увеличивается на валидационных, значит, модель переобучается (излишне подстраивается под тренировочные данные) и теряет способность к обобщению на неизвестные для неё примеры.

На рисунке 3 показано, как менялась ошибка модели в течение первой эпохи на тренировочных и валидационных данных. Как видно, обучение происходило корректно и переобучения не происходило.

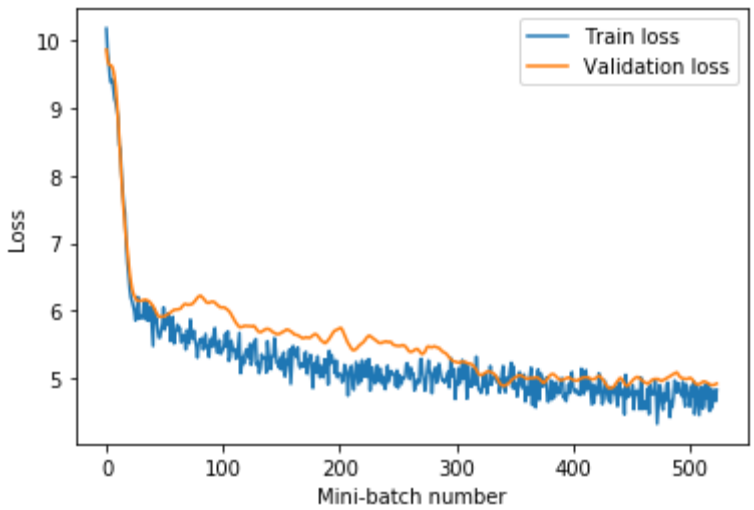

Рисунок *3*. – Ошибка модели на тренировочных и валидационных данных

Полученные результаты

Для проверки работы модели было взято простое предложение на английском языке “I have an apple and two oranges”. Ниже для каждого слова представлены предсказанные моделью вероятности того, что это слово входит в ответ на некоторый вопрос.

0. I - 0.13

- Have - 0.12
- An - 0.17
- Apple - 0.21
- And - 0.17
- Two - 0.24
- Oranges - 0.21
- . - 0.01

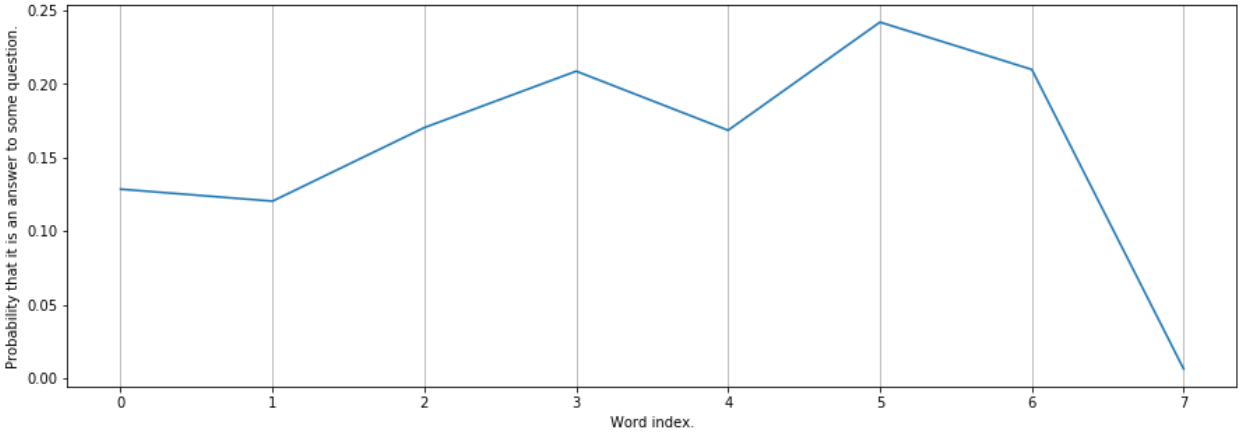

Рисунок *4.* – Вероятности того, что каждое слово входит в ответ

Далее были выбраны наиболее вероятные сочетания слов и отправлены на вход второй модели. Вопросы, полученные для выбранных сочетаний.

- Для сочетания “An apple” сгенерирован вопрос “What is the name of the <UNK>?”
- Для слова “two” сгенерирован вопрос “How many children?”

Как видно, вопросы корректны с точки зрения грамматики. Модель отличает, например, числительные от существительных и подбирает в зависимости от этого правильную форму вопроса. Однако контекст вопроса может не совпадать с контекстом начального предложения. Также есть проблема с тем, что в некоторых случаях модель предсказывает неизвестное для словаря слово <UNK> как наиболее вероятное. Это связано с небольшим размером словаря.

Результат может быть улучшен, если продолжить обучать модель в течение большего количества эпох и на большем словаре, для чего может понадобиться мощное аппаратное обеспечение. Также можно развивать модель в сторону усложнения ее архитектуры.

Заключение

На основании проведенных исследований известной модели [4, с. 190] генерации вопросов был разработан алгоритм, почти максимально учитывающий лексические и синтаксические связи внутри предложения. В результате удалось получить пару значений - вопрос, корректно составленный на основе языковых правил, и ответ на него.

Список литературы

[1.] Using Natural Language Processing for Smart Question Generation [Электронный ресурс]. — Режим доступа: https://software.intel.com/en-us/articles/using-natural-language-processing-for-smart-question-generation. — Дата доступа: 20.01.2020.

[2.] Himanshu Jethwani, Mohd Shahid Husain, Mohd Akbar. Automatic Question Generation from Text. — International Journal for Innovations in Engineering, Science and Management, 2015, УДК 004.8.

[3.] Николенко С., Кадурин А., Архангельская Е.. Глубокое обучение. — СПБ.: Питер, 2019, с. 251, УДК 004.8.

[4.] David Foster. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play 1st Edition, УДК 004.8.

NEURAL NETWORK FOR AUTOMATED QUESTION GENERATION

D. A. Klebanov
Student of BSUIR

K. I. Akulich
Student of BSUIR

M. A. Kalugina
Associate Professor of Informatics Department of the BSUIR

U. U. Astrashab
Student of BSUIR

D. S. Sovpel
Student of BSUIR

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
E-mail: u.astraszab@gmail.com, dmitriy.klebanov@gmail.com, sovspace@gmail.com, kirillakulch0@gmail.com, marina_kalugina@list.ru.

Abstract. An algorithm for automated question generation is presented. Included topics are main approaches to the problem, model architecture and implementation, results of work and further perspectives on development of the model.

Keywords: question generation, big data, machine learning, neural networks, natural language processing.