Министерство науки и высшего образования
Российской Федерации
«Московский физико-технический институт
(государственный университет)»
Физтех-школа прикладной математики и информатики
Факультет управления и прикладной математики
Кафедра прикладных проблем теоретической и
математической физики

Сайранов Данил

Отбор релевантных предложений в задаче построения вопросно-ответных систем

03.03.01 — Прикладные математика и физика

Выпускная квалификационная работа
(бакалаврская диссертация)

Научный руководитель:
к.ф-м.н.
Серебряков Владимир Алексеевич

Москва
2018

Оглавление

1. Введение 3
1.1. Определения и вводимые понятия 3
1.2. Обзор литературы 5
2. Постановка задачи 6
3. Решение 7
3.1. Построение генератора ответов на вопрос 7
3.2. Ранжирование предложений в документах относительно вопросов 10
4. Эксперименты 13
4.1. Данные 13
4.2. Генератор ответов на вопрос 15
4.3. Ранжирование предложений в документе 18
4.4. Результаты работы R-Net 19
5. Заключение 19

Список литературы 21

Аннотация

В данной работе исследуются методы выделения релевантных предложений в задаче построения вопросно-ответных систем. В качестве рассматривамых методов выступают метод построения генератора ответов на вопрос и метод ранжирования предложений из текста относительно вопроса. Данные алгоритмы позиционируются как внешние модули предобработки данных, которые могут быть встроены в любую из реализованных на сегодняшний день вопросно-ответных систем.

1. Введение

Извлечение информации из текстов на сегодняшний день является актуальной задачей. Одним из способов решения этой задачи выступают вопросно-ответные системы, которые сильно шагнули вперед за последнее десятилетие и достигли достаточно хороших результатов. Имея некоторое множество документов, такие системы пытаются найти в нем ответ на вопрос, сформулированный на естественном языке. Однако в большинстве реальных задач документы представляют собой десятки, а иногда и сотни предложений, где ответ на вопрос хранится лишь в одном из них. В таких ситуациях часто падает точность ответа, что мотивирует исследователей модернизировать архитектуру систем, добавлять различные модули, изменять формат представления данных и прочее.

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Пример документа и вопросов по нему

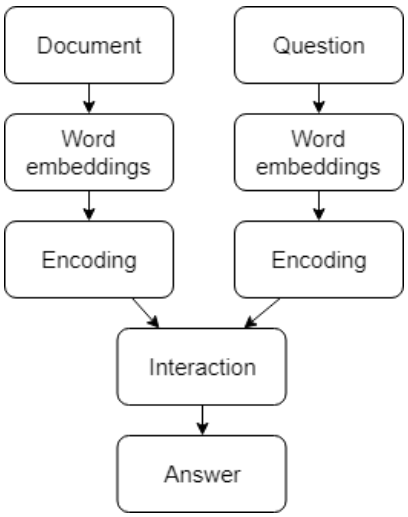
Целью данной работы является изучение способов выделения предложений, которые связаны с заданным вопросом и, более того, могут содержать в себе ответ. Таким образом вопросно-ответные системы будут искать ответ не во всем документе, а только в выделенной части.

1.1. Определения и вводимые понятия

Определим вопросно-ответные системы как системы, на вход которых подается пара (D, q), где D – документ, q – вопрос по документу, а на выход возвращается участок из документа, являющийся ответом на вопрос.

Этапы работы вопросно-ответных систем.

Общая архитектура нейросетевых вопросно-ответных систем представляет собой 5 этапов работы [1]. На первом этапе на вход подается пара (D, q). Вторым этапом каждому слову w_i^D документа и каждому слову вопроса w_j^q в соответствие ставится их векторное представление $\mathbf{w}_i^D$ и $\mathbf{w}_j^q$ соответственно (*Embedding*). На третьем этапе происходит построение контексто-зависимых векторов для каждого вектора $\mathbf{w}_i^D$ и $\mathbf{w}_j^q$ (*Encoding*). Контекстно-зависимые вектора хранят некоторую информацию об окружающем каждое слово тексте, что позволяет учитывать такие явления, как, например, контекстную синонимию. Далее на четвертом этапе происходит обогащение полученных векторов документа информацией о вопросе (*Interaction*). Чаще всего для этого реализуется механизм внимания [2], позволяющий нейронной сети присваивать больший вес той информации из документа, которая более важна относительно данного вопроса. Последним, пятым, этапом работы является этап вывода ответа на вопрос исходя из полученной на предыдущих этапах информации.



Архитектура нейросетевых вопросно-ответных систем

Типом вопроса (ответа) будем называть ту информацию, которая в нем запрашивается (содержится). Другими словами, если в вопросе (для ответа аналогично) содержится информация о времени, в которое произошло то или иное событие, то вопрос имеет тип «дата». Если же содержится информация о каком-либо человеке, то типом вопроса будет являться «личность». Аналогично можно определить и другие типы.

В качестве примеров можно рассмотреть следующее:

- **Вопрос:** «В каком году родился Пушкин?»

Ответ: «Пушкин родился 6 июня 1799 г.»

В данном случае типом вопроса и типом ответа является «дата».

- **Вопрос:** «Что такое паровая машина?»

  Ответ: «Первая паровая машина построена Дени Папеном.»

  В этом примере типом вопроса является «определение», но типом ответа - «личность».

Очевидно, что необходимым условием корректности пары «вопрос-ответ» является то, что тип ответа должен совпадать с типом вопроса.

Языковая модель - распределение вероятностей на последовательностях слов. Иными словами, пусть дана последовательность слов длины m, тогда языковая модель ставит ей в соответствие вероятность $P(w_1, \cdots, w_m)$ того, что такая последовательность может встретиться в данном языке. Заметим, что такую вероятность можно записать следующим образом:

$$\mathbf{P}(w_1, \cdots, w_m) = \prod_{i=1}^{m} \mathbf{P}(w_i | w_1, \cdots, w_{i-1})$$

Условная вероятность $\mathbf{P}(w_i | w_1, \cdots, w_{i-1})$ показывает вероятность того, что следующим словом в тексте $w_1, \cdots, w_{i-1}$ будет слово w_i.

1.2. Обзор литературы

Отметим, что изначально вопросно-ответные системы были неким набором простых правил извлечения информации из коллекции неструктурированных документов [3, 4]. Однако точность таких систем была достаточно низкой и в связи с активным развитием информационных технологий набора правил становилось недостаточно. На сегодняшний день архитектуры вопросно-ответных систем стали гораздно сложнее. Появление инструментов анализа семантики естественных языков, основанных на векторном представлении слов и дистрибутивной семантике, таких как [5, 6], а так же

появление достаточно больших выборок позволило применить нейросетевые архитектуры и добиться с их помощью хороших результатов в задаче построения вопросно-ответных систем [1, 7, 8]. Однако на достаточно больших документах точность таких систем может упасть, что мотивирует исследовать различные методы уменьшения текстов.

Для построения генератора ответов на вопрос в работе используется инструментарий Open-NMT, подробно описанный в [9]. Для сравнения предложений в работе используются ранжирующая модель DRMM [10], реализованная в пакете [11] и так же векторные представления предложений, описанные в [12].

2. Постановка задачи

С увеличением размера документов увеличивается сложность поиска ответа на вопрос в них. Современные вопросно-ответные системы не могут хранить и обрабатывать достаточно большие документы, что мотивирует исследовать способы уменьшения их размеров таким образом, чтобы уменьшенные документы все еще хранили в себе ответ на вопрос.

Пусть имеется некоторое множество неразмеченных документов D и некоторое множество вопросов Q.

Задача. Предложить алгоритмы выделения k предложений документе $d_i \in D$ на основе вопроса $q_j \in Q$ таких, что среди них найдется ответ на вопрос, в задаче построения вопросно-ответных систем.

При исследованиии методов выделения релеватных предложений использовалась выборка SQuAD, которая содержит в себе более 500 статей из англоязычной Википедии и более 100000 пар «вопрос-ответ».

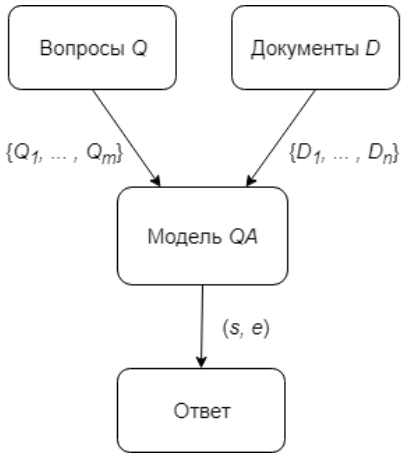


Схема изначальной системы

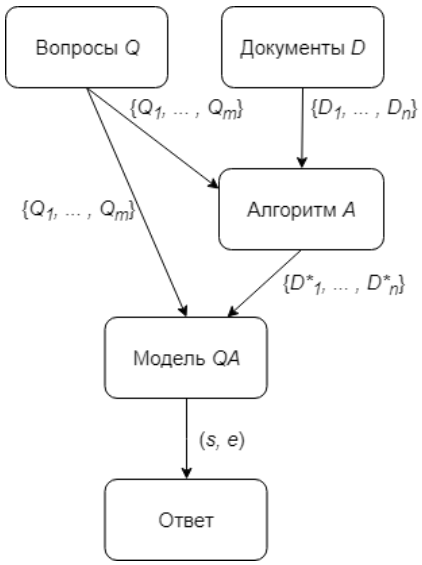


Схема модифицированной системы

3. Решение

В работе рассмотрены два метода выделения релевантных вопросу предложений:

- Метод построения генератора ответа на вопрос и ранжирования предложений документа относительно сгенерированного предложения,
- Метод ранжирования предложений в документе относительно вопроса.

3.1. Построение генератора ответов на вопрос

Гипотеза. Ответ на вопрос, сформулированный на естественном языке, схож с предложением из текста, содержащим ответ, по типу.

Построение генератора.

Так как вопросы по документам содержат в себе различное количество слов, то генератор ответов должен быть способным обрабатывать входные последовательности переменной длины. Для таких задач отлично подходят рекуррентные нейронные сети, которые также показывают хорошие

результаты в задачах генерации текста, что показано в работах [13, 14, 15].

Задачей генератора ответов на вопрос является построение языковой модели. Пусть задано множество пар $D = (x_i, y_i)$, где x_i - входная последовательность, y_i - ожидаемая выходная последовательность $\forall i = \overline{1, N}$. Функцией потерь при генерации ответа в данном случае является функция:

$$Q(D) = -\sum_{(x,y)\in D} \log \mathbf{P}(y|x)$$

Заменив вероятность под логарифмом произведением условных вероятностей получим:

$$Q(D) = -\sum_{(x,y)\in D} \log \prod_{i=1}^{|y|} \mathbf{P}(w_i^y | x, w_1^y, \cdots, w_{i-1}^y) =$$

$$= -\sum_{(x,y)\in D} \sum_{i=1}^{|y|} \log \mathbf{P}(w_i^y | x, w_1^y, \cdots, w_{i-1}^y) = \sum_{(x,y)\in D} H(x, y),$$

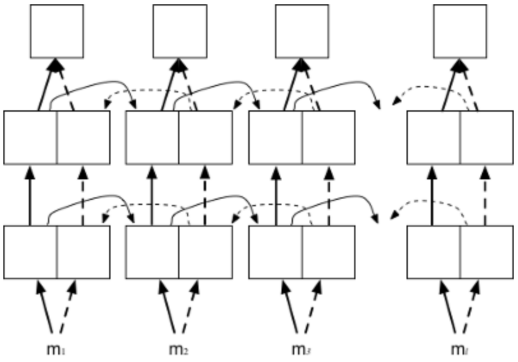
где w_i^y - i-й элемент последовательности y.

Для вычисления значения функции потерь на одном предложении $H(x, y) = -\sum_{i=1}^{|y|} \log \mathbf{P}(w_i^y | x, w_1^y, \cdots, w_{i-1}^y)$ необходима модель, генерирующая слова последовательно по одному. Такие модели называются моделями типа seq2seq [16] и способны выводить последовательности переменной длины, в зависимости от входных данных.

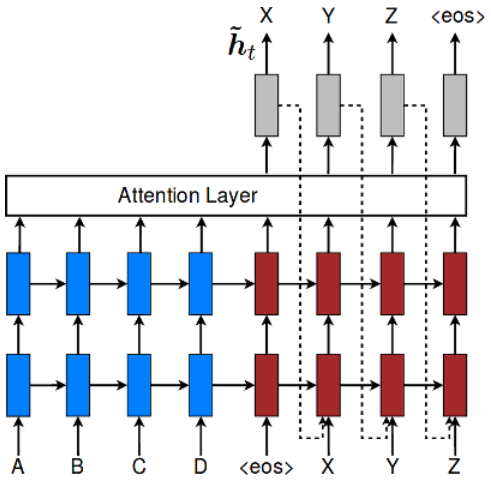
Таким образом, для генератора ответов на вопрос естественно выбрать модель seq2seq компонентами которой являеются encoder и decoder. Компонента модели encoder в такой архитектуре позволяет вычислить векторное представление $\mathbf{s}$ входной последовательности, а компонента decoder позволяет генерировать слова последовательно по одному, тем самым представляя функцию $H(\mathbf{s}, y)$ в виде суммы логарифмов условных вероятностей.

Так же стоит отметить, что использование механизма внимания улучшает качество работы нейросетевых моделей на многих задачах, например, машинном переводе [17].

В качестве архитектуры компоненты encoder выбрана двунаправленная рекуррентная нейронная сеть. Такая сеть в отличие от однонаправленной запоминает контекст не только с левой стороны, но и с правой, что позволяет ей достигать лучших результатов на достаточно длинных последовательностях.



Архитектура двунаправленной нейронной сети



Архитектура генератора

В качестве векторного представления слов выбраны широко используемые предобученные векторы GloVe [6].

Алгоритм отбора релевантных предложений представлен ниже.

Algorithm 1 Отбор релевантных предложений

procedure SELECTSENTENCES(q, D, k)
    preprocess D
    preprocess q
    $answer$ = generateAnswer(q)
    for sentence s in D **do**
        $scores(s)$ = inferSent($answer, s$)
    $selected$ = topK($scores, k$) ▷ returns top-K sentences
    return $selected$

Процедура preprocess понижает регистр текстовых данных. После этого на основе вопроса q генерируется ответ $answer$. Далее, для каждого

предложения s из документа D рассчитывается его близость $scores(s)$ к сгенерированному ответу $answer$ с помощью inferSent. После чего с помощью функции topK, возвращающей k предложений с наибольшим значением их схожести со сгенерированным предложением, возвращаются k наиболее релевантных предложений. Параметр k задается вручную.

3.2. Ранжирование предложений в документах относительно вопросов

Пусть имеется вопрос q и документ $D = \{d_1, \cdots, d_n\}$, где d_i – i-е предложение в документе.

Модель TF-IDF. Для определения релевантности предложений вопросу построим векторную модель предложений документа с помощью меры TF-IDF.

Для каждого слова из вопроса q посчитаем его частоту вхождений в предложение d_i как отношение отношение числа вхождений в d_i к общему числу слов в d_i:

$$TF(t, d_i) = \frac{n_t}{\sum\limits_k n_k},$$

где n_t - количество вхождений слова t в предложение d_i.

Обратная частота слова из вопроса q определяется как инверсия частоты, с которой это слово встречается в предложениях документа D:

$$IDF(t, D) = \ln \frac{|D|}{|\{d \in D | \, t \in d\}|}$$

Таким образом для каждого термина t в документе можно вычислить его меру TF-IDF:

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

Пусть $T_q = \{t_1, \cdots, t_m\}$ - множество слов, содержащихся в вопросе q. Тогда каждому предложению $d_i \in D$ и вопросу q можно поставить в соответствие

вектора:

$$\mathbf{d_i} = TF\text{-}IDF(T_q, d_i, D)$$

$$\mathbf{q} = TF\text{-}IDF(T_q, q, q)$$

Степень релевантности предложения вопросу оценивается следующим образом:

$$similarity(q, d_i) = \cos\theta = \frac{\langle \mathbf{q}, \mathbf{d_i} \rangle}{\|\mathbf{q}\|_2 \cdot \|\mathbf{d_i}\|_2},$$

где угол θ - угол между векторами $\mathbf{d_i}$ и $\mathbf{q}$.

Модель InferSent. Модель Infersent[12] - нейронная сеть, сопоставляющая предложению его векторное представление в некотором заданном пространстве. Каждому предложению $d_i \in D$ и вопросу q ставится в соответствие вектора:

$$\mathbf{d_i} = InferSent(d_i)$$

$$\mathbf{q} = InferSent(q)$$

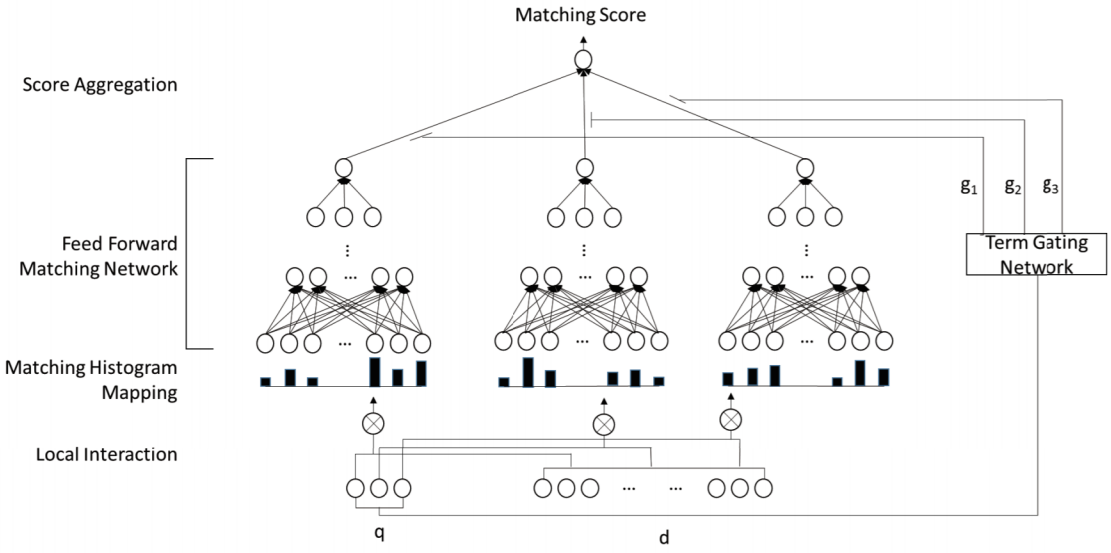
Далее, аналогично модели TF-IDF, степень релевантности предложения вопросу оценивается с помощью косинусной меры:

$$similarity(q, d_i) = \cos\theta = \frac{\langle \mathbf{q}, \mathbf{d_i} \rangle}{\|\mathbf{q}\|_2 \cdot \|\mathbf{d_i}\|_2}$$

Модель DRMM.

DRMM (Deep Relevance Matching Model) - ранжирующая модель определяющая релевантность предложения запросу, основываясь на трех факторах: точное совпадение, важность слова в вопросе, длина предложения. Более подробно модель рассмотрена в работе [10].

Пусть вопрос и предложение из документа представлено в виде набора векторов $q = \{w_1^q, \cdots, w_m^q\}$ и $d = \{w_1^d, \cdots, w_n^d\}$. Тогда степень релевантно-



Архитектура DRMM

сти предложения вопросу s рассчитывается следующим образом:

$$z_i^0 = h(w_i^q \otimes d),\ i = \overline{1,n}$$

$$z_i^l = tanh(W^l z_i^{l-1} + b^l),\ i = \overline{1,m},\ l = \overline{1,L}$$

$$s = \sum_{i=1}^{m} g_i z_i^L$$

где $\otimes$ - некоторый оператор взаимодействия слова из вопроса со словом из предложения в документе, функция h - отображение локальных взаимодействий в гистограмму релевантностей слов предложения слову из вопроса, z_i^l при $l = \overline{0,L}$ - значения на промежуточных скрытых слоях для i-го слова из вопроса, g_i при $i = \overline{0,m}$ - вектор весов для каждого из слов в вопросе, вычисляемый компонентой Term Gating Network. Матрица W^l - l-я матрица весов нейросети, b^l - l-й вектор смещения нейросети.

Алгоритм отбора релевантных предложений в таком случае будет выглядить следующим образом:

Аналогично предыдущему алгоритму, сначала происходит понижение регистра входных текстовых данных (preprocess). Далее для каждого предложения s в документе D рассчитывается его схожесть $scores(s)$ с вопросом q с помощью функции model. Функция model является обобщением трех

Algorithm 2 Отбор релевантных предложений

```
1: procedure SelectSentences(q, D, k)
2:     preprocess D
3:     preprocess q
4:     for sentence s in D do
5:         scores(s) = model(q, s)
6:     selected = topK(scores, k)          ▷ returns top-K sentences
7:     return selected
```

моделей, указанных выше, другими словами, при выполнении алгоритма эта фукнция должна быть заменена на одну из трех моделей: TF-IDF, inferSent, DRMM. После чего, возвращаются k предложений с наибольшим значением схожести.

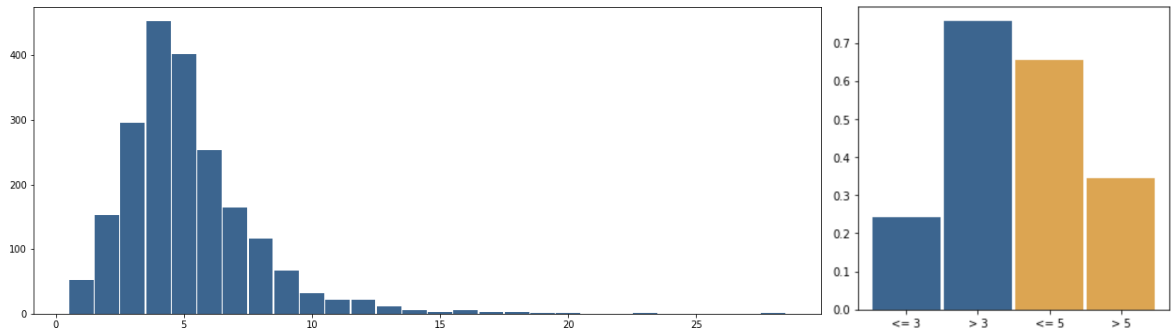
4. Эксперименты

4.1. Данные

В качестве данных, на которых оценивалась работа алгоритмов, взята выборка SQuAD, представляющая собой более ста тысяч вопросов по множеству отрывков из статей англоязычной Википедии. Ответом на каждый вопрос из выборки является участок текста из соответствующего отрывка.

Для обучения ранжирующей модели используется обучающая выборка SQuAD. На гистограммах ниже изображено распределение документов в зависимости от количества предложений в них. Как можно заметить на гистограмме слева, среднее количество предложений в документах приблизительно равно 4,5. Большая часть выборки (около 65% от всех документов) содержит в себе не более 5 предложений, что можно увидеть на гистограмме справа.

В качестве данных для обучения генератора ответов на вопрос были



Слева: Распределение количества документов относительно количества предложений в них в выборке SQuAD. **Справа:** Доля документов в выборке SQuAD с определенным условием на количество предложений.

взяты и предобработаны три выборки: SQuAD [18], MS Marco [19], SelQA [20]. Выборки SQuAD и SelQA не содержат сформулированных ответов на вопросы, поэтому в них в качестве ответа выбраны предложения, в которых содержатся участки текста, содержащие ответ. Выборка MS Marco в отличие от предыдущих содержит в себе хорошо сформулированные ответы на вопросы, что гипотетически должно положительно повлиять на качество работы генератора. Примеры пар «вопрос – ответ» можно увидеть ниже.

- MS Marco:

    1. **What is the Meiji emperors name?**

       - The Meiji emperor's name is Mutsuhito.

    2. **What does subside mean?**

       - Subside means to sink to a low or lower level.

- SQuAD:

    1. **When was levi's stadium awarded the right to host super bowl 50?**

       - On may 21, 2013, nfl owners at their spring meetings in boston voted and awarded the game to levi's stadium.

    2. **What is the term for a task that generally lends itself to**

being solved by a computer?

- A **computational problem** is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.

- SelQA:

1. **In what year was the chilean national museum of fine arts built?**

   - Museums in chile such as the chilean national museum of fine arts built in 1880 feature works by chilean artists.

2. **Which Academy Award did Kevin Spacey win for his work on The Usual Suspects?**

   - Christopher McQuarrie was nominated for the Best Original Screenplay and Kevin Spacey was nominated for Best Supporting Actor at the Academy Awards.

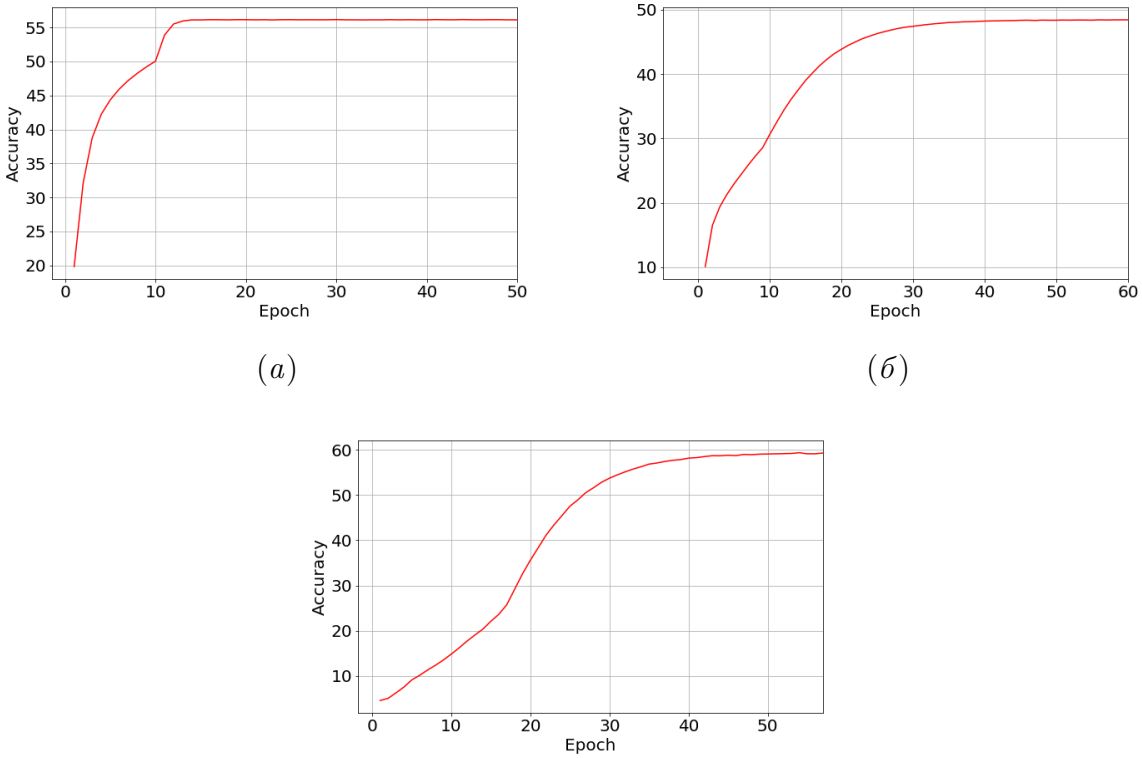
Исходя из примеров, видно, что в ответах выборок SQuAD и SelQA так же содержится лишняя «шумовая» информация, которая снижает качество обучения генератора.

4.2. Генератор ответов на вопрос

При реализации генератора ответов на вопрос в работе использовался инструментарий Open-NMT [9], в котором реализованы модели с seq2seq архитектурой для машинного перевода, в частности, с двунаправленной рекуррентной сетью в качестве компоненты encoder и механизмом внимания. Графики обучения генератора на различных выборка можно увидеть

ниже. При обучении был задействован механизм early stopping, для предотвращения переобучения сети. На графиках видно, что после определенной эпохи точность работы сети на тестовых выборках меняется незначительно. Поэтому для экспериментов были выбраны:

- эпоха 13 на выборке MS Marco
- эпоха 42 на выборке SQuAD
- эпоха 50 на выборке SelQA



(*а*) (*б*)

(*в*)

Графики обучения генератора на выборках: **(а)** MS Marco, **(б)** SQuAD, **(в)** SelQA

Примеры предложений, сгенерированных обученным на выборке MS Marco алгоритмом:

- **When is groundhog day?**

  - A groundhog day is on may 28, 2016.

- **What is the smallest us state in terms of land area?**

  - Alaska is the smallest us state in terms of land area.

- **What is nightwatch?**

  - Nightwatch is a type of nightwatch what what and what

- **Definition of sacrament?**

  - The definition of sacrament is definition

- **Should you eat before or after a workout?**

  - You should eat before a workout.

Степень схожести сгенерированного предложения с предложениями из документа оценивалась с помощью ранжирующей модели Infersent, описанной выше. После чего отбирались k-наиболее близких пар. Для оценки качества отбора релевантных предложений было рассчитано значение

$$Recall@k = \frac{tp}{total},$$

где tp – число вхождений предложений, содержащих ответ, в k-наилучших предложений, $total$ – общее число предложений, содержащих ответ.

Выборка	Recall@1	Recall@3	Recall@5
SQuAD	0.567	0.888	0.971
MS Marco	0.660	0.918	0.980
SelQA	0.278	0.703	0.910

Таблица 1. Качество работы генератора ответов, обученного на различных выборках

Исходя из результатов, приведенных в таблице 1, можно сделать вывод, что гипотеза о том, что сгенерированные ответы по типу схожи с предложениями в документах, содержащих ответ, подтвердилась. Проведенные

на выборке SQuAD эксперименты показали, что при выделении трех наиболее близких к сгенерированному ответу предложений, достигается точность 91.8% того, что среди выделенных предложений содержится ответ, что является достаточно хорошим показателем.

4.3. Ранжирование предложений в документе

Используемая модель DRMM была взята из пакета для сравнения текстов MatchZoo[11]. В качестве обучающей выборки модель принимает на вход тройки (r, s_1, s_2), где s_1 и s_2 – сравниваемые предложения, r – коэффициент релевантности предложений друг другу, поэтому для обучения модели был сформирован набор троек для каждого документа D такой, что s_1 пробегает все множество вопросов Q по документу D, s_2 для любого $s_1 \in Q$ пробегает по всем предложениям из D и r принимает значение 1, если s_2 содержит ответ на вопрос s_1, или значение 0 в противном случае.

Аналогично предыдущему, для оценки качества отбора релевантных предложений было посчитано значение $Recall@k$ для каждой из трех моделей.

Модель	Recall@1	Recall@3	Recall@5
TF-IDF	0.627	0.882	0.962
InferSent	0.682	0.920	0.982
MatchZoo DRMM	0.423	0.510	0.561

Таблица 2. Качество ранжирования предложений в документе относительно вопроса с помощью различных моделей

Результаты данного эксперимента (приведены в таблице 2) показали, что можно достигнуть большей точности, сравнивая каждое предложение документа с вопросом, в отличие от сравнения предложений со сгенерированным ответом на вопрос.

4.4. Результаты работы R-Net

Для оценки качества работы алгоритмов в контексте задачи построения вопросно-ответных систем были проведены эксперименты, где в качестве вопросно-ответной системы взята система R-Net.

	Топ-1		Топ-3		Топ-5	
Алгоритм	EM	F1	EM	F1	EM	F1
R-Net	**70.918**	**79.644**	**70.918**	**79.644**	**70.918**	**79.644**
R-Net + MS Marco	52.030	59.360	65.124	74.157	68.547	77.693
R-Net + SQuAD	51.012	57.423	62.364	71.177	65.529	74.011
R-Net + SelQA	26.564	32.331	44.455	52.261	59.101	64.395
R-Net + TF-IDF	53.623	60.935	63.917	72.472	69.833	78.625
R-Net + InferSent	50.674	56.146	62.081	70.707	64.967	75.039
R-Net + DRMM	54.408	62.189	62.668	71.180	64.428	73.045

Таблица 3. Качество работы R-Net с различными алгоритмами

Результаты эксперимента, приведенные в таблице 3, показали, что все алгоритмы понижают качество работы вопросно-ответной системы. Однако, стоит отметить, что использование алгоритмов при отборе 3 или более релевантных предложений в документе, незначительно влияют на качество работы вопросно-ответной системы, позволяя тем самым уменьшить количество предложений в документах.

5. Заключение

В рамках проведенного исследования были построены алгоритмы выделения релевантных предложений в задаче построения вопросно-ответных систем. Было предложено два алгоритма: ранжирование предложений в документе относительно ответов на вопрос, сформированных генератором

ответов, и ранжирование предложений в документе относительно самих вопросов.

Были достигнуты достаточно хорошие результаты при выделении предложений в документе - в 90% случаев выделенные предложения содержали в себе ответ на вопрос. Однако добавление алгоритмов выделения предложений в качестве модуля предобработки данных для вопросно-ответных систем понизило качество работы вопросно-ответной системы R-Net на 4-8%. Стоит отметить, что такое понижение качества работы в задачах с достаточно большими документами пренебрежимо, т.к. такие алгоритмы могут помочь обойти существующие ограничения на объем входных данных вопросно-ответных систем.

Список литературы

1. FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension / H.-Y. Huang, C. Zhu, Y. Shen [и др.] // ArXiv e-prints.
2. Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate // ArXiv e-prints. 2014-09.
3. Álvaro Rodrigo, Pérez-iglesias Joaquín, Peñas Anselmo [и др.]. A Question Answering System based on Information Retrieval and Validation.
4. Clark Peter, Thompson John, Porter Bruce. A Knowledge-Based Approach to Question-Answering. 1999. 05.
5. Efficient Estimation of Word Representations in Vector Space / T. Mikolov, K. Chen, G. Corrado [и др.] // ArXiv e-prints.
6. Pennington Jeffrey, Socher Richard, Manning Christopher D. GloVe: Global Vectors for Word Representation // Empirical Methods in Natural Language Processing (EMNLP). 2014. С. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.
7. Weissenborn D., Wiese G., Seiffe L. Making Neural QA as Simple as Possible but not Simpler // ArXiv e-prints.
8. Bidirectional Attention Flow for Machine Comprehension / M. Seo, A. Kembhavi, A. Farhadi [и др.] // ArXiv e-prints.
9. OpenNMT: Open-Source Toolkit for Neural Machine Translation / G. Klein, Y. Kim, Y. Deng [и др.] // ArXiv e-prints.
10. A Deep Relevance Matching Model for Ad-hoc Retrieval / J. Guo, Y. Fan, Q. Ai [и др.] // ArXiv e-prints.
11. MatchZoo: A Toolkit for Deep Text Matching / Yixing Fan, Liang Pang, JianPeng Hou [и др.] // arXiv preprint arXiv:1707.07270. 2017.
12. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data / A. Conneau, D. Kiela, H. Schwenk [и др.] // ArXiv e-prints.

13. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation / Y. Wu, M. Schuster, Z. Chen [и др.] // ArXiv e-prints. 2016-09.
14. Breaking the Softmax Bottleneck: A High-Rank RNN Language Model / Z. Yang, Z. Dai, R. Salakhutdinov [и др.] // ArXiv e-prints. 2017-11.
15. Merity S., Shirish Keskar N., Socher R. Regularizing and Optimizing LSTM Language Models // ArXiv e-prints. 2017-08.
16. Sutskever I., Vinyals O., Le Q. V. Sequence to Sequence Learning with Neural Networks // ArXiv e-prints. 2014-09.
17. Luong M.-T., Pham H., Manning C. D. Effective Approaches to Attention-based Neural Machine Translation // ArXiv e-prints.
18. SQuAD: 100,000+ Questions for Machine Comprehension of Text / P. Rajpurkar, J. Zhang, K. Lopyrev [и др.] // ArXiv e-prints.
19. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset / T. Nguyen, M. Rosenberg, X. Song [и др.] // ArXiv e-prints.
20. Jurczyk T., Zhai M., Choi J. D. SelQA: A New Benchmark for Selection-based Question Answering // ArXiv e-prints.