Сравнение качества end-to-end обучаемых моделей в задаче ответа на вопросы в диалоге с учетом контекста

Агафонов А. М, Рякин И. С., Хохлов И. Ю., Литвиненко В. В., Великовский Н. А., Ануфриенко О. Д.

frizman04@gmail.com, ryakin.is@phystech.edu, khokhlov.iyu@gmail.com, vladimir.litvinenko.1997@gmail.com, velik.nikita@gmail.com, oleg.anufriyenko@phystech.edu

Moscow Institute of Physics and Technology

В работе рассматривается вопросно-ответная система (QA). Задан фрагмент текста и несколько последовательных вопросов. Ответы на первые N вопросов известны. Нужно сформировать ответ на N+1 вопрос и указать непрерывный промежуток в тексте заданного фрагмента текста. Исследование проводится на новых данных , для которых на данный момент имеется только базовый алгоритм. В работе изучается возможность улучшения этого базового алгоритма. Для этого предлагается изучить существующие механизмы учета контекста (k-ctx, append, etc) и исследовать возможность их добавления в другие модели (R-NET, DrQA), либо предложить собственные для повышения качества по мере F1. Для изучения поведения модели используется attention visualization, обучаемых эмбеддингов, а также анализ ошибочных ответов.

Ключевые слова: *контекст, machine comprehension, вопросно-ответная система (QA), нейросеть, FLOW mechanism .*

1 Введение

В работе решается задача ответа на вопросы с учетом контекста и учетом хода диалога. Используется датасет QuAC [1]. Он содержит 14K QA диалогов(100K пар вопрос-ответ). Датасет представляет собой диалог между учителем и учеником. Считается, что учитель знает какую-то информацию, например статью с Wikipedia, а ученик, в свою очередь, задает вопросы по этой статье. Вопросы могут быть с множественным выбором, без ответа, основанные на диалоге и т.д. В данном разделе в первую очередь представлены обзоры на релевантные алгоритмы данной тематики, решающие данную задачу на различных данных. Примерами являются подходы [2–5]

1.1

В работе Bi-Directional Attention Flow for Machine Comprehension [2] используется датасет SQuAD [6]. Решается задача получения ответа на вопрос с учетом контекста. В отличие от других алгоритмов в этой работе используется современный алгоритм Bi-Directional Attention Flow. Обычно с помощью Attention [7] определяют небольшую часть контекста и сосредотачиваются на ней. В статье [2] описывается представление контекста в многоуровневой иерархии.

1.2

В работе Reading Wikipedia to Answer Open-Domain Questions [3] рассматривается задача ответа на вопросы-факты по открытым источникам данных на примере Wikipedia. Для обучения модели использовались Wikipedia, SQuAD [6], CuratedTREC, WebQuestions, WikiMovies. Алгоритм состоит из двух этапов. Первый по заданному вопросу ищет статью (статьи), в которой далее требуется искать ответ. Реализовано при помощи TF-IDF

меры применительно к биграммам (использовано хеширование для ускорения). Второй по заданному вопросу ищет в статье, найденной в первом пункте, промежуток, содержащий нужный ответ. В работе можно выделить такие интересные особенности как то, что помимо GloVe embedding [8] в векторе признаков для слова добавляется еще 3 бинарных признака наличия конкретного токена в вопросе (полностью совпадает, совпадает в нижнем регистре, совпадает в нормальной форме) и часть речи, тип именованной сущности, частота появления в параграфе.

1.3

Наиболее релевантный алгоритм для используемого в нашем исследовании датасета на сегодняшний день - FlowQA [5]. В статье предложено решение учета предыдущих вопросов и ответов: отвечая на N-ый вопрос, алгоритм опирается не только на изначальный контекст, но и на предыдущие N-1 вопрос-ответы, - контекст разговора постоянно обновляется. Это позволяет отвечать на такие простые вопросы как “Где?” и “Когда?”, исходя из информации, которая была представлена ранее. Процедура получения ответа выполняется в три шага. Во-первых, вопрос и контекст кодируются с помощью GloVe embedding [8]. Во-вторых, полученные вектора подаются на вход LSTM [9] с использованием Attention [7]. В-третьих, полученные вектора обрабатываются с целью получения вероятности для слова быть началом/концом ответа на поставленный вопрос.

С целью сравнения результатов полученных методом, предложенным в этой работе, и результатов, представленных в QuAC [1], проведен вычислительный эксперимент на исходных данных QuAC [1].

2 Описание алгоритма

В этом разделе мы опишем подход к обучению системы ответов на вопросы с контекстом. На первом этапе на основе контекста из статьей ивзлекается самый близкий параграф, который далее передается в вопросно-ответную модель. Далее вопросно-ответная модель находит в тексте параграфа начало и конец ответа.

2.1 Выбор параграфа

Наша модель выбора параграфа выбирает несколько параграфов с самым маленьким TF-IDF косинусным расстоянием от вопроса.

2.2 Описание модели

Наша модель состоит из следующих слоев:

Эмбеддинги: Мы векторизуем слова при помощи предобученных эмбеддингов. Мы также векторизуем отдельные символы в 20-мерные обучаемые векторы, которые после проходят через CNN и Max-pooling. Далле мы производим конкатенацию эмбеддингов слов и символов и передаем их на следующий слой.

Предобработка: Далее используется двунаправленная GRU [10] для преобразования эмбеддингов в контекстно-ориентированные эмбеддинги.

Attention: Двунаправленная модель attention из модели BiDAF [2] используется для представления контекста с учетом запроса. Обозначим ветор контекстного слова под номером i как h_i, а j-ый вектор вопроса как q_j и длину вопроса и контекста n_q и n_c соответственно. Опишем attention между h_i и q_j следующим образом:

$$a_{ij} = w_1 \cdot h_i + w_2 \cdot q_j + w_3 \cdot (h_i \odot q_j)$$

где w_1, w_2 и w_3 обучаемые параметры. После этого получаем новые вектора контекста c_i:

$$p_{ij} = \frac{e^{a_{ij}}}{\sum\limits_{j'=1}^{n_q} e^{a_{ij'}}}$$

$$c_i = \sum_{j=1}^{n_q} q_j p_{ij}$$

Также мы рассчитываем вектор запроса-контекста q_c:

$$m_i = \max_{1 \leqslant j \leqslant n_q} a_{ij}$$

$$p_i = \frac{e^{m_i}}{\sum\limits_{i=1}^{n_c} e^{m_i}}$$

$$q_c = \sum_{i=1}^{n_c} h_i p_i$$

Финальный вектор, полученный для каждого токена, собирается путем конкатенации $h_i, c_i, h_i, h_i \odot c_i$ и $q_c \odot c_i$. Полученный вектор мы далее пропускаем через линейный слой и нелинейную активацию ReLU.

Self-Attention: Следующим шагом мы используем residual слой Self-attention [11]. Входной вектор пропускается через двунаправленную GRU [10]. Далее мы применяем точно такой же механизм attention, но теперь с самим собой. В данном случае мы не используем вектор запроса-контекста (q_c) и выставляем $a_{if} = -inf$, если $i = j$.

Как и ранее, пропускаем сконкатенированный вектор через линейный слой и ReLU. Так как данный слой является residual, то мы складываем его вход и выход.

Предсказание: На последнем шаге используется двунаправленная GRU [10] и линейный слой, которые возвращают score начала ответа для каждого токена. Внутренние состояния конкатенируются с входными значениями, пропускаются через еще одну двунаправленную GRU и линейный слой, возвращая score конца ответа для каждого токена.

Все полученые score мы пропускаем через слой Softmax и получаем $P_{start}(i)$ - вероятность, что i-й токен является началом ответа и $P_{end}(j)$ - вероятность, что j-й токен является концом ответа. Важно отметить, что сглаживание происходит по всем параграфам, полученным на первом этапе алгоритма. За правильный ответ мы принимаем:

$$(i_{start}, j_{end}) = \underset{(i,j)}{argmax}(P_{start}(i) + P_{end}(j) | i \leqslant j \leqslant i + 17)$$

Dropout: Мы также использовали Dropout, который случайным образом обращал в ноль некоторые внутренние состояния на протяжении всего обучения. Dropout применялся ко всем GRU [10], эмбеддингам слов, а также ко всем входам механизмов attention с частотой 0.2.

3 Эксперимент

Для обучения нашей модели мы выбрали выборку QuAC [1]. Датасет содержит 98,407 пар вопрос-ответ из 13,594 диалогов. Каждый диалог содержит от 4 до 12 вопросов. Диалоги были произведены на 3,611 уникальных статьях из Википедии. Реализованная нами

Таблица 1 F1 мера моделей и человека на QuAC.

Модель	F1
`BiDAF++`	60.1
`FlowQA`	64.1
`Наше решение`	63.7
`Человек`	80.8

версия алгоритма обучалась батчами размера 45 с оптимизатором Adadelta [12]. В реализации алгоритма использовались 300-мерные GloVe [8] эмбеддинги слов. Мы использовали рамерность 100 для каждой GRU [10] и 200 для каждого линейного слоя после механизма attention. Во время обучения мы поддерживали экспоненциальную скользящую среднюю весов со скоростью затухания 0,999. Во время тестирования мы использовали усредненные веса. Наша модель обучалась на 80% выборки в течение 100 эпох. Полученный результат можно наблюдать на таблице 1.

4 Заключение

В работе была предложена модель, базирующаяся на решении BiDAF [2], для решения задачи ответа на вопросы в диалоге с учетом контекста. Данная модель обучалась на актуальной выборке QuAC [1]. С целью сравнения качества ответов модели с уже существующими решениями был проведен эксперимент, демонстрирующий повышенное качество ответов по сравнению с базовым алгоритмом BiDAF [2]. Хотя наш подход обеспечивает значительный прирост производительности, все еще есть возможности для улучшения. В будущем мы хотели бы исследовать еще более эффективные методы нахождения ответов на вопросы с учетом контекста, основываясь на решении FlowQA [5].

Литература

[1] *Eunsol Choi, He He, Wen-tau Yih, Yejin Choi, Mohit Iyyer, Mark Yatskar, Percy Liang, Luke Zettlemoyer* QuAC : Question Answering in Context // arXiv preprint arXiv:1808.07036, 2018

[2] *Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hananneh Hajishirzi* Bi-Directional Attention Flow for Machine Comprehension // ICLR, 2017

[3] *Danqi Chen, Adam Fisch, Jason Weston, Antoine Bordes* Reading Wikipedia to Answer Open-Domain Questions // arXiv preprint arXiv:1704.00051, 2017

[4] *Natural Language Computing Group, Microsoft Research Asia* R-NET: Machine Reading Comprehension with Self-Matching Networks // `https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf`, 2017

[5] *Hsin-Yuan Huang, Eunsol Choi, Wen-tau Yih* FlowQA: Grasping Flow in History for Conversational Machine Comprehension // arXiv preprint arXiv:1810.06683, 2018

[6] *Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang* SQuAD: 100,000+ questions for machine comprehension of text // Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2016

[7] *Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria* Recent Trends in Deep Learning Based Natural Language Processing // arXiv preprint arXiv:1708.02709v8, 2018

[8] *Jeffrey Pennington, Richard Socher, Christopher D. Manning* GloVe: Global Vectors for Word Representation // `https://nlp.stanford.edu/pubs/glove.pdf`, 2014

[9] *Sepp Hochreiter and Jurgen Schmidhuber* Long short-term memory // Neural Computation, 9(8): 1735–1780, 1997

[10] *Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio* Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation // arXiv preprint arXiv:1406.1078v3, 2014

[11] *Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, Chengqi Zhang* DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding // arXiv preprint arXiv:1709.04696, 2017

[12] *Matthew D. Zeiler* ADADELTA: An Adaptive Learning Rate Method // arXiv preprint arXiv:1212.5701, 2012