УДК 004.912

МНОГОЯЗЫЧНАЯ МОДЕЛЬ ОТВЕТА НА ВОПРОСЫ С ОТКРЫТОЙ ПРЕДМЕТНОЙ ОБЛАСТЬЮ

Сыромятников Н.В., Цуркан А.В.
проф. каф. СПО, к.т.н. Рувинская В.М.
Одесский Национальный Политехнический Университет, УКРАИНА

АННОТАЦИЯ. Рассматривается разработка многоязычной модели ответа на вопрос с открытой предметной областью. Для реализации были задействованы глубинные нейронные сети на основе архитектуры *Transformer,* методы градиентного бустинга и технологии векторного представления слов.

Введение. В последние годы сфера NLP стремительно развивается и соизмеримо расширяется спектр решаемых задач. Доказательством этому может послужить задача нахождения ответа на вопрос с открытой предметной областью (ODQA), суть которой состоит в нахождении фрагмента текста, релевантного вопросу. Источником информации при этом может служить любое хранилище документов, например Википедия, как это было в случае с DrQA [1].

Цель работы. Целью работы является создание многоязычной модели ODQA, заточенной под русский и украинский языки. Важным аспектом реализации также является повышение скорости обработки запросов путем уменьшения размеров компонентов модели без существенных потерь в качестве.

Основная часть работы. В общем случае архитектура модели нахождения ответа на вопрос состоит из двух модулей: ранжировщика и чтеца [2]. Задачей ранжировщика является сортировка документов в порядке их релевантности запросу. Чтец же находит в документе фрагмент, который с наибольшей вероятностью является ответом на заданный вопрос.

Рассмотрим реализацию ранжировщика. Стандартным подходом является сравнение сходства двух строк. Для этой задачи существуют технологии векторного представления, такие как Word2Vec (использует слова) и FastText (использует n-граммы). Однако для многоязычных моделей они не применимы. В связи с этим Facebook AI Research разработали архитектуру для использования многоязычных эмбеддингов [3], представленную в библиотеке Laser.

Для оптимизации решения под нашу задачу воспользуемся классификатором, на вход которому будет подаваться горизонтальная конкатенация векторных представлений двух фраз, полученных с помощью Laser, а на выходе будем получать вероятности предсказания схожести. Сам классификатор представляет из себя модель градиентного бустинга, реализованную в библиотеке XGBoost. Используемые параметры модели: n_estimators=1500, lr=0.1, max_depth=3, subsample=0.8. Итоговые метрики моделей классификации близости предоставлены в таблице 1.

Помимо ансамбля деревьев решения (Laser-XGBoost) была рассмотрена модель на основе архитектуры Transformer. BERT – модель представления естественного языка, которая во всех слоях использует левый и правый контекст [4]. Данная модель после *fine-tuning* показывает *state-of-the-art* на многих задачах классификации текстов. В задаче определения сходства двух текстов воспользуемся bert-base-multilingual-cased и добавим выходным слоем линейный классификатор с последующей *sigmoid* в качестве функции активации, функция потерь – *cross-entropy loss*.

В результате оценивания Bert-classifier оказался сравним с моделью Laser-XGBoost на тестовых сетах русского и украинского языков, но в то же время он значительно медленнее. Для ускорения воспользуемся *knowledge distillation* [5]. Данный метод заключается в передаче знаний от “учительской” сети (Bert classifier) “ученику” (DistilBert-classifier), в нашем случае это 4-слойный bert-cased. Формула 1 отображает функцию потерь сети-ученика во время обучения:

$$L = \alpha * L_{cross\ entropy}(\hat{y}_{student}, y_{student}) + (1-\alpha) * L_{mse}(\hat{y}_{student}, \hat{y}_{teacher}) \quad (1)$$

где α – гиперпараметр в пределах [0, 1], во время обучения было использовано значение 0.5.

Таблица 1 – Сравнение моделей в задаче классификации схожести

Модель/метрики	F1	F1 ru	F1 uk	F1 en	Время, с 100 док-ов	Время, с 1000 док-ов
Laser-XGBoost-classifier	**0.915**	0.92	0.931	**0.895**	0.015	0.063
Bert-classifier	0.875	**0.93**	0.917	0.784	36.2	370
DistilBert-classifier	0.865	0.928	**0.944**	0.732	12	123.1

Laser-XGBoost обучена на датасете из 190 тысяч семплов запросов и контекстов на трёх языках, Bert-модели – на его сокращенной (50 тысяч семплов) версии. DistilBert по скорости превзошла Bert в 3 раза, потеряв в точности только на английском языке.

В таблице 2 модели сравниваются на задаче ранжирования. Здесь на вход, помимо запроса, поступает несколько контекстов, и при использовании базовых моделей recall будет ощутимо снижаться. Для решения этой проблемы вводится параметр W – ширина окна. W = n определяет, что вычисление результатов будет проводиться по n-первым документам, отсортированным по близости к запросу. Вероятность для последовательных моделей считалась как их среднее.

Таблица 2 – Сравнительные характеристики моделей в задаче ранжирования контекстов

Модель/метрики	F1	Precision	Recall	F1 ru	F1 uk	F1 en	Время, с 100 док-ов	Время, с 1000 док-ов
Laser-XGBoost (W=20)	0.906	**0.995**	0.832	0.931	0.901	0.895	0.015	0.063
Laser-XGBoost (W=50)	**0.945**	0.977	**0.915**	**0.965**	**0.945**	**0.929**	0.015	0.063
Laser-XGBoost (W=50) → DistilBert (W=5)	0.925	0.99	0.868	0.955	0.937	0.868	6.1	6.2
Laser-XGBoost (W=50) → DistilBert (W=10)	**0.943**	0.99	0.9	**0.964**	**0.944**	0.914	6.1	6.2

Лучшей стала модель Laser-XGBoost с окном в 50 позиций, но наиболее целесообразно использовать последовательную Laser-XGBoost (W=50) → DistilBert (W=10), потому как она сокращает до 10 позиций результирующий набор, что повысит производительность модуля QA.

Заключительным этапом является разработка модели чтеца. На данный момент *state-of-the-art* результаты в этой задаче показывают модели на основе BERT. Воспользуемся предобученной версией BertForQA от Hugging Face и дообучим её на аугментированном датасете, состоящем из SQuAD 2.0 и SDSJ 2017. Результаты тестирования предоставлены в таблице 3.

Таблица 3 – Результаты обучения модели QA

Модель/метрики	EM	F1	EM uk	F1 uk	Время, с 10 док-ов	Время, с 1000 док-ов
BertForQA	0.63	0.71	0.74	0.823	0.225	2.35

EM – *exact match*, точное совпадение ответа с предсказанным фрагментом. Измерение для украинского языка производилось на отдельном сете из 100 вопросов, собранном вручную [6].

Итоговая архитектурная схема модели ODQA представлена на рисунке 1.

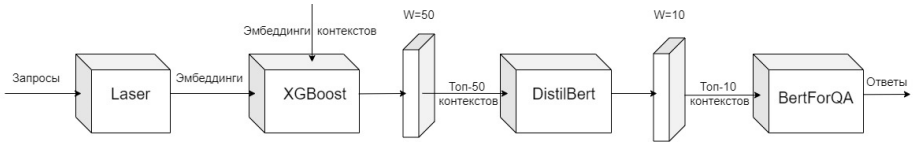

Рис. 1 – Итоговая схема модели ODQA, W – ширина окна

Выводы. В результате сравнительного анализа технологий была предложена и реализована архитектура многоязычной модели для решения задачи ответа на вопрос с открытой предметной областью. F1 ранжировщика для русского и украинского языков составила 96.4 и 94.4 единиц соответственно, при этом без существенных потерь в качестве после ускорения модели Bert в 3 раза и уменьшения размера окна выдачи (W) в 5 раз. F1 модели ответа на вопрос в виде взвешенного среднего на символьном уровне составила 71 единицу, *exact match* – 63 единицы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архив электронных публикаций научных статей [Электронный ресурс]. – Режим доступа: URL: https://arxiv.org/abs/1704.00051. – Reading Wikipedia to Answer Open-Domain Questions.
2. Архив электронных публикаций научных статей [Электронный ресурс]. – Режим доступа: URL: https://arxiv.org/abs/1709.00023. – R3: Reinforced Reader-Ranker for Open-Domain Question Answering.
3. Архив электронных публикаций научных статей [Электронный ресурс]. – Режим доступа: URL: https://arxiv.org/abs/1812.10464. – Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond.
4. Архив электронных публикаций научных статей [Электронный ресурс]. – Режим доступа: URL: https://arxiv.org/abs/1810.04805. – Pre-training of Deep Bidirectional Transformers for Language Understanding
5. Архив электронных публикаций научных статей [Электронный ресурс]. – Режим доступа: URL: https://arxiv.org/abs/1903.12136. – Distilling Task-Specific Knowledge from BERT into Simple Neural Networks.
6. Dev-human-v2.0 [Электронный ресурс]. – Режим доступа: URL: https://github.com/s-e-r-g-y/context-based-qa-for-uk/blob/master/datasets-uk/dev-human-v2.0.json. – Назва з екрана.