В архиве 3 папки, в каждом свое решение: panorama, solution_augmentations, solution. Финальное решение лежит в папке solution. Ниже приводится описание каждой из папок

Для выполнения задачи прочитал публикации на тему Fake News Detection. Выделил несколько статей:

- Cross-lingual Evidence Improves Monolingual Fake News Detection в этой статье авторы используют поиск заголовков в гугл на пяти различных языках и оценивают косинусную близость найденных текстов с исходным. Дополнительно используют AlexaRank для оценки посещаемости сайта и доверия к источнику.
- FakeBERT: Fake news detection in social media with a BERT-based deep learning approach авторы сравнивают подходы от классических до трансформеров и предлагают брать после Bert не линейный слой над CLS, а несколько слоев сверток над ембеддингами каждого токена
- Transformer based Automatic COVID-19 Fake News Detection System в этой статье авторы сравнили несколько подходов к решению проблемы и оценивали их относительно метрики F1. Лучшим стал ансамбль из трансформеров

Panorama

Решение с использованием cross-lingual evidence значительно повышало метрику для всех существующих подходов, поэтому первой интересно было применить ее. Но в результате тестирования выяснилось, что для предоставленного набора данных можно обойтись и без сравнения статей между собой, достаточно определить есть ли среди первых 10 выдач поисковой системы сайты с доменом "panorama", "ryb.ru", "kolibri.press". Написал парсер, проверил его на всем тесте, результатом стал 1 False Negative для вирусного заголовка “'Рамзана Кадырова выдвинули на Нобелевскую премию мира”

В папках Test и Train находятся скрипты для парсинга заголовков, в ноутбуке kontur-panorama.ipynb находится код исследования.
Конечно, отправлять такой подход слишком скучно, (хотя зачем использовать ml, когда можно его не использовать). Поэтому продолжил делать задание дальше. После просмотра данных определил, что данных очень мало, большинство заголовков содержат меньше 15 слов. Поэтому очевидным был подход с увеличением датасета.

Solution_augmentations

(К сожалению, не проходит по правилам, но в task.md это указано совсем неявно)

Для поиска новых фейковых заголовков загрузил дамп телеграм канала панорамы (папка telegram_parsing) и вытащил оттуда уникальные заголовки (panorama-parsing.ipynb). Для

реальных заголовков взял датасет ленты с kaggle и собрал случайные статьи за 2018-2019 год (lenta-news.ipynb). Для каждого нового заголовка проверял его на вхождение в предложенный датасет через значение edit distance. Далее, разделил полученные данные на train, valid, test в соотношении 90, 5, 5 (merge-parsed-data.ipynb) и для train использовал модель MT-5 large из модуля Russian Paraphrasers для получения перефразированных заголовков (paraphrase.ipynb).
Дополнительно к этому взял Fake News Dataset со статьями на английском языке и выбрал наиболее чистые категории (fake-news-eng.ipynb). Данные на английском языке собрал из предположения, что на верхних слоях трансформеры становятся независимы к входному языку обучаясь на корпусах мультиязычного текста.
Далее, исходя из того, что решения на трансформерах набирали лучший скор согласно исследованиям, и не было ограничений к работе модели, то решил остановиться на данной архитектуре, пробовал решения основанные на T5 и RoBerta. Сначала использовал модель T5 base с оставленным русским и английским словарем, далее взял ruT5 base обученную преимущественно на русском языке, она показала результат чуть выше. Затем взял енкодер RuRoberta-large, он показал лучший результат как на валидации так и на тестовых данных.
Код с обучением и валидацией моделей лежит в папке models

Модель	Eval micro F1	Test micro F1
T5 first eng then ru	0.911	0.906
T5-first-eng-then-ru-aug	0.915	0.894
ruT5 plain train	0.933	0.921
ruT5	0.935	0.93
ruRoberta	0.946	0.94

Можно заметить, что ruT5 с файнтюном на английском языке показала больший скор чем без него.

Solution

Так как тренировочные данные нельзя менять, то для финального сабмита была выбрана модель RuRoberta-large потому что она показала наилучший результат на тестовой выборке при обучении на большем датасете. Было произведено обучение на предложенных тренировочных данных с помощью 5 фолдов и усреднения предсказаний с каждого фолда для уменьшения дисперсии. Результат работы predictions.tsv лежит в корне архива. Ноутбук ruroberta-5-folds лежит в папке Solution, также его можно запустить через kaggle