Софийски университет „Св. Кл. Охридски”

Факултет по математика и информатика

Курсов Проект

по ИД “Откриване на знания в текст“

на тема:

„Класификация на новини чрез методите, базирани на трансформатори, BERT, roBERTa и DistilBERT”

Студент: Надежда Францева

Ф.Н.: 8MI3400357

Курс: „ИИ“

Учебна година: 2023/2024

Преподаватели: **ст. н.с. I ст. д-р Преслав Наков,**

ас. Димитър Димитров

Декларация за липса плагиатство:

- Плагиатство е да използваш, идеи, мнение или работа на друг, като претендираш, че са твои. Това е форма на преписване.
- Тази курсова работа е моя, като всички изречения, илюстрации и програми от други хора са изрично цитирани.
- Тази курсова работа или нейна версия не са представени в друг университет или друга учебна институция.
- Разбирам, че ако се установи плагиатство в работата ми ще получа оценка “Слаб”.

05.07.2024 г. Подпис на студента:

Съдържание

1. Увод

1.1 Мотивация

В съвременния свят извличането на информация от големи обеми текстови данни е от решаващо значение за различни приложения като медии, финанси, и други. Необходимостта от автоматизирано класифициране на новини може да подобри процесите на анализ и вземане на решения, като осигури по-ефективно и точно разпределение на информацията.

1.2 Цел на проекта

Целта на текущия курсов проект е да изследва и сравни различни модели за класификация на новини, използвайки различни модели на база трансформатори, като BERT, RoBERTa и DistilBERT. Проектът цели да демонстрира възможностите на съвременни технологии за обработка на естествен език в приложения за класификация на текстове.

1.3 Задачи на проекта

1. **Извличане и обработка на данни:** Зареждане, обединение и обработка на текстови данни от набора от данни **News Classification - Inshort daily news data**
2. **Използване на модели на база трансформатори:**
    - Интеграция и обучение на BERT, RoBERTa и DistilBERT модели за класификация на текстови данни.
    - Оценка на точността и сравнение с традиционните модели.
3. **Сравнителен анализ и заключения:**
    - Сравнение на различните модели за класификация на новини спрямо точността и други ключови метрики.
    - Изводи за предимствата и недостатъците на всяка методология.
4. **Дискусия и по-нататъшно развитие:**
    - Обсъждане на резултатите и възможности за подобрение на моделите.
    - Предложения за бъдещи изследвания и разширения на проекта.

2. Преглед на областта

Бързият напредък в областта на обработката на естествен език (Natural Language Processing, NLP) води до значително подобрение на резултатите в различни задачи, включително класификацията на текстове. В последните години се наблюдава нарастващ интерес към използването на трансформаторни модели като ключов инструмент за подобряване на точността в различни NLP приложения.

2.1 Подходи и методи за класификация на текстове

В контекста на задачата за класификация на текстове, традиционните методи като машинно самообучение са били широко използвани. Сред тях се включват SVM (машина с поддържащи вектори), Naive Bayes и RandomForest, които предлагат добри резултати в различни сценарии, но често са ограничени от необходимостта от ръчно инженерство на характеристиките и ограничена способност да улавят сложни взаимодействия в текста.

2.2 Използване на трансформаторни модели

С настъпването на трансферното обучение (transfer learning), трансформаторните модели като BERT (Bidirectional Encoder Representations from Transformers), RoBERTa (Robustly Optimized BERT Pretraining Approach) и DistilBERT (Distill Bidirectional Encoder Representations from Transformers) са се установили като state-of-the-art решения в областта на NLP. Тези модели са способни да улавят семантични и синтактични зависимости в текста благодарение на способността си да обработват контекстуална информация и да използват големи количества данни за предварително обучение.

2.3 Сравнителен анализ на трансформаторни модели

Сравнителни изследвания показват, че трансформаторните модели като BERT, RoBERTa и DistilBERT демонстрират значително по-висока точност в сравнение с традиционните подходи към класификацията на текстове. Тяхната способност да улавят сложни зависимости и контекстуална информация води до значително подобрение на резултатите в различни NLP задачи, включително и в задачи за класификация на новини.

3. Проектиране

Проектирането на системата за класификация на новини може да бъде описано в няколко основни стъпки, всяка от които включва определени компоненти и задачи. Ще разгледаме архитектурата на системата и ще опишем всяка част от процеса.

Основни Компоненти и Архитектура

1. **Зареждане и Обединяване на Данни**
    - **Описание**: Зарежда множество CSV файлове и ги обединява в единен DataFrame.
    - **Инструменти и технологии**: pandas, glob.
2. **Предварителна Обработка на Данни**
    - **Описание**: Разделяне на текстовете и категориите, преобразуване на категориите в числови етикети и разделяне на данните на обучителен и тестов набор.
    - **Инструменти и технологии**: pandas, sklearn.
3. **Невронни Мрежи и Трансформъри**
    - **Описание**: Обучение на модели на базата на предварително обучени трансформъри (BERT, RoBERTa, DistilBERT) за последователна класификация.
    - **Инструменти и технологии**: transformers, tensorflow.

4. Реализация, тестване/експерименти

4.1 Използвани технологии, платформи и библиотеки

Технологии и платформи:

- **Google Colab:** Платформа за изпълнение на Python код в облака, която предоставя достъп до мощни изчислителни ресурси като GPU, необходими за обучение на големи модели.
- **TensorFlow:** Популярна библиотека за машинно самообучение и дълбоки невронни мрежи, използвана за изграждане и обучение на трансформаторни модели като BERT, RoBERTa и DistilBERT.

Библиотеки:

- **Transformers:** Библиотека от Hugging Face, която предоставя предварително обучени модели за обработка на естествен език, включително BERT, RoBERTa и DistilBERT. Тази библиотека улеснява зареждането, обучение и оценка на модели за NLP задачи.
- **Pandas:** Библиотека за работа с данни в Python, която осигурява лесни за използване структури и аналитични инструменти за работа с таблични данни.
- **TfidfVectorizer:** Част от scikit-learn, използвана за преобразуване на текстовете в числови вектори, базирани на TF-IDF (Term Frequency-Inverse Document Frequency).

Избор на средствата и начин за използването им:

Google Colab е избран заради лесния достъп до мощни изчислителни ресурси и интеграцията с Google Drive, което улеснява управлението на данните и моделите.

Python е предпочитан поради богатата екосистема от библиотеки за машинно самообучение и обработка на естествен език.

TensorFlow и Transformers са използвани заради тяхната способност да работят с трансформаторни модели, които показват висока точност при задачи за класификация на текстове.

Pandas е основният инструмент за обработка на данни, който позволява ефективна манипулация на големи масиви от данни и тяхната подготовка за моделиране.

TfidfVectorizer е използван за преобразуване на текстовите данни в числови представяния, подходящи за традиционните модели за машинно самообучение.

4.2 Реализация/Провеждане на експерименти

Проектът е реализиран под формата на *Colab Notebook*:

https://colab.research.google.com/drive/1MGP1OPWxr7vPqHwfETktCJGMBj2dd6rp

Като са използвани следните данни от *Kaggle*:

https://www.kaggle.com/datasets/kishanyadav/inshort-news

Основните използвани метрики за оценка на резултатите са точност (accuracy), прецизност (precision), recall (покриваемост), претеглен F1-score и подкрепа (support).

Разглеждаме оценките на всеки класификатор за задачата, използвайки предоставените метрики:

BERT модел:

```
Epoch 1/3
606/606 [==============================] - 369s 499ms/step - loss: 1.9459 - accuracy: 0.1539 - val_loss: 1.9459 - val_accuracy: 0.1708
Epoch 2/3
606/606 [==============================] - 294s 485ms/step - loss: 1.9459 - accuracy: 0.1490 - val_loss: 1.9459 - val_accuracy: 0.1708
Epoch 3/3
606/606 [==============================] - 312s 516ms/step - loss: 1.9459 - accuracy: 0.1540 - val_loss: 1.9459 - val_accuracy: 0.1708
152/152 [==============================] - 23s 149ms/step - loss: 1.9459 - accuracy: 0.1708
BERT accuracy: 0.17079207301139832
```

```
152/152 [==============================] - 23s 153ms/step
BERT F1 Score: 0.2918207005856817
BERT Precision: 0.35116249502305535
BERT Recall: 0.3655115511551155
              precision    recall  f1-score   support

           0       0.69      0.14      0.23       387
           1       0.34      0.93      0.49       414
           2       0.30      0.34      0.32       383
           3       0.38      0.70      0.50       298
           4       0.53      0.27      0.36       401
           5       0.00      0.00      0.00       259
           6       0.00      0.00      0.00       282

    accuracy                           0.37      2424
   macro avg       0.32      0.34      0.27      2424
weighted avg       0.35      0.37      0.29      2424
```

Табл.4 Резултати от BERT

- **Точност (accuracy)**: 0.1708
- **F1 Score**: 0.2918
- **Прецизност (precision)**: 0.3512
- **Покриваемост (recall)**: 0.3655

Анализ на резултатите на BERT:

- **Прецизност и покриваемост**: Ниски стойности на прецизността (precision) и покриваемостта (recall) показват затруднение в правилното класифициране на данните от модела BERT. Например, за клас 5 и клас 6 прецизността и покриваемостта са нулеви, което означава, че моделът не предсказва успешно тези класове.
- **F1 Score**: Също така е нисък, което указва на недостатъчност в балансирането между прецизност и покриваемост за различните класове.

DistilBERT модел:

```
Epoch 1/5
606/606 [==============================] - 186s 238ms/step - loss: 0.3910 - accuracy: 0.9042 - val_loss: 0.2020 - val_accuracy: 0.9402
Epoch 2/5
606/606 [==============================] - 140s 232ms/step - loss: 0.1654 - accuracy: 0.9493 - val_loss: 0.1841 - val_accuracy: 0.9476
Epoch 3/5
606/606 [==============================] - 141s 233ms/step - loss: 0.1333 - accuracy: 0.9563 - val_loss: 0.1760 - val_accuracy: 0.9472
Epoch 4/5
606/606 [==============================] - 141s 232ms/step - loss: 0.1144 - accuracy: 0.9583 - val_loss: 0.1990 - val_accuracy: 0.9352
Epoch 5/5
606/606 [==============================] - 141s 232ms/step - loss: 0.0994 - accuracy: 0.9616 - val_loss: 0.1684 - val_accuracy: 0.9509
152/152 [==============================] - 11s 75ms/step - loss: 0.1684 - accuracy: 0.9509
DistilBERT accuracy: 0.9509075880050659
152/152 [==============================] - 16s 72ms/step
DistilBERT F1 Score: 0.9506883180598212
DistilBERT Precision: 0.951024629890535
DistilBERT Recall: 0.9509075907590759
              precision    recall  f1-score   support

           0       0.89      0.88      0.88       362
           1       0.96      0.99      0.97       382
           2       0.96      0.90      0.93       404
           3       0.99      0.99      0.99       308
           4       0.97      0.96      0.97       398
           5       0.93      0.98      0.96       274
           6       0.95      0.97      0.96       296

    accuracy                           0.95      2424
   macro avg       0.95      0.95      0.95      2424
weighted avg       0.95      0.95      0.95      2424
```

Табл.5 Резултати от DistilBERT

- **Точност (accuracy)**: 0.9509
- **F1 Score**: 0.9507
- **Прецизност (precision)**: 0.9510
- **Покриваемост (recall)**: 0.9509

Анализ на резултатите на DistilBERT:

- **Прецизност и покриваемост**: Високи стойности на прецизността и покриваемостта за всички класове, което показва, че моделът успешно класифицира данните във всички категории.

- **F1 Score**: Също така е висок, което оказва добро балансиране между прецизността и покриваемостта, необходимо за класификацията на различните класове.

5. Заключение

DistilBERT постига значително по-добри резултати в сравнение с BERT в тази конкретна задача. Той демонстрира висока прецизност, покриваемост и F1-Score за всички класове, което го прави предпочитан избор за класификационни задачи в сравнение с по-тежките и ресурсоемки модели като BERT и RoBERTa.

Идеи за по-нататъшно развитие, усъвършенстване или други експерименти:

1. **Оптимизация на моделите**:
    - Изследване на техники за оптимизация на трансформаторните модели, за да се намали изчислителната сложност и времето за обучение.
2. **Разширяване на набора от данни**:
    - Използване на по-големи и по-разнообразни набори от данни за подобряване на обобщаващата способност на моделите.
    - Включване на нови категории и увеличаване на броя на статиите във всяка категория.
3. **Фина настройка на трансформаторните модели**:
    - Провеждане на по-дълбока фина настройка на трансформаторните модели с помощта на специфични за задачата данни, за да се подобри тяхната производителност.
    - Изследване на използването на различни трансформаторни модели и сравняването им с BERT и RoBERTa.
4. **Обработка на текстове на различни езици**:
    - Разширяване на проекта за включване на текстове на различни езици чрез използване на многоезични трансформаторни модели като mBERT и XLM-R.

6. Използвана литература

[1] Support Vector Machines - Scikit-learn, https://scikit-learn.org/stable/modules/svm.html

[2] Naive Bayes - Scikit-learn, https://scikit-learn.org/stable/modules/naive_bayes.html

[3] Random Forest Classifier - Scikit-learn, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[4] Classify text with BERT - TensorFlow, https://www.tensorflow.org/text/tutorials/classify_text_with_bert

[5] RoBERTa - Transformers, https://huggingface.co/docs/transformers/model_doc/roberta

[6] DistilBERT - Transformers, https://huggingface.co/docs/transformers/en/model_doc/distilbert