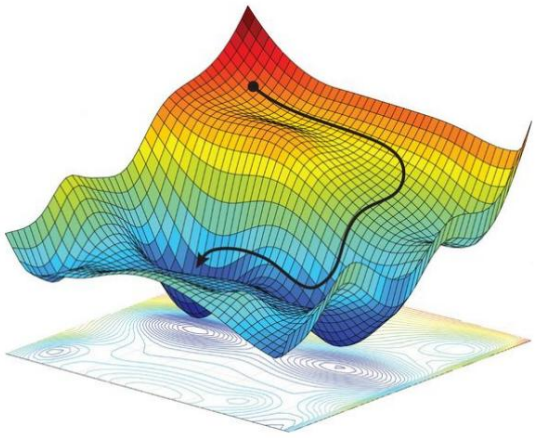
Fakulta elektrotechniky a informatiky



Факультет электротехники и компьютерных наук

Кафедра кибернетики и искусственного интеллекта

Тема: Нейронные сети

курс 2022/2023

Использование нейронных сетей для обнаружения плагиата

Обработано:
Орыдорога Богдан, Ройко Алексей,
Цимбота Владислав и Зеленская Злата

KKUI
Katedra kybernetiky a umelej inteligencie

Содержание

1. Введение

Плагиат — это использование чужой работы или идей без надлежащего указания первоисточника. Это считается проступком, который может иметь серьезные последствия в академической и профессиональной среде. Плагиат может включать в себя копирование текста, изображений, музыки или другого имущества без разрешения и представление его как вашей собственной работы.

Плагиат тесно связан с авторскими правами и правами на литературную собственность. Авторское право защищает права создателей и владельцев оригинальных произведений, таких как книги, статьи, музыка и программное обеспечение. Когда кто-то занимается плагиатом, он фактически нарушает эти права, используя произведение без согласия владельца и без надлежащего указания авторства.

Например, студент может заняться плагиатом, скопировав и вставив абзац из опубликованной статьи в свое сочинение без ссылки на источник. Точно так же писатель может заняться плагиатом, включив части чужого романа в свою работу без разрешения или указания авторства. В обоих случаях плагиат нарушает права правообладателя и может привести к юридическим последствиям.

В истории было несколько известных примеров плагиата, например, в случае изобретателя и ученого Николы Теслы. Тесла известен своим значительным вкладом в разработку электрических систем переменного тока, которые составляют основу современного распределения электроэнергии. Однако он столкнулся с жесткой конкуренцией со стороны других изобретателей, включая Томаса Эдисона, который был сторонником односторонних электрических систем[1].

Работы Теслы часто затмевались работами других изобретателей, и его вклад не всегда получал признание. В 1888 году Тесла получил несколько патентов на свой асинхронный двигатель переменного тока, что в то время было прорывом в электротехнике. Однако не Тесла прославился этим изобретением, а Джордж Вестингауз, американский бизнесмен и инженер, купивший патенты Теслы и коммерциализировавший технологию[1]. Хотя Тесла в конце концов получил признание за свою работу, затмение его изобретений другими изобретателями и бизнесменами является примером того, как плагиат и отсутствие надлежащей атрибуции могут повлиять на самые блестящие умы.

В последние годы технологии играют важную роль в обнаружении и предотвращении плагиата.
Одним из наиболее перспективных подходов в этой области является применение нейронных сетей для выявления и анализа потенциально плагиатного контента. В следующих разделах мы рассмотрим концепцию нейронных сетей, их применение для обнаружения плагиата и то, как они могут помочь повысить точность и эффективность выявления плагиата.

2 Спецификация области применения и описание проблемы

Использование нейронных сетей для обнаружения плагиата дает несколько преимуществ и делает их привлекательным выбором для этой задачи. Ниже приведены некоторые ключевые аргументы, подкрепленные ссылками на исследования:

- Высокая точность — нейронные сети, особенно модели глубокого обучения, оказались очень точными при выявлении закономерностей и корреляций в больших файлах данных. Исследования продемонстрировали «эффективность» моделей нейронных сетей в выявлении сходств и различий в тексте, что обеспечивает надежный и надежный метод обнаружения плагиата[2] .

- Понимание семантики – нейронные сети, особенно те, которые основаны на трансформирующих архитектурах, таких как BERT, продемонстрировали большой успех в понимании эмантического значения текста[3]. Эта способность позволяет им обнаруживать не только точные совпадения, но и перефразированный или переписанный контент, что является ключом к эффективному обнаружению плагиата.

- Адаптивность — нейронные сети способны обучаться и адаптироваться к новым данным и изменяться в соответствии с языковыми тенденциями[4]. Эта адаптивность делает их подходящими для задачи обнаружения плагиата, поскольку их можно научить распознавать различные формы и стили письма и адаптироваться к изменениям языковых стандартов.

- Скалер 'ness' - обнаружение плагиата часто требует обработки большого количества текста. Нейронные сети могут быть распараллелены и реализованы на графических процессорах (GPU), что позволяет эффективно обрабатывать большие наборы данных[5].

Таким образом, использование нейронных сетей для обнаружения плагиата подтверждается различными исследованиями, которые доказывают их высокую точность, семантическое понимание, адаптивность и «сплавляемость», что делает их хорошим выбором для решения этой задачи. Помимо способности адаптироваться к новым образцам и стилям текста, нейронные сети могут быстро и автоматически сканировать большой объем текста, что позволяет выявлять случаи плагиата в режиме реального времени. Ожидается, что с прогрессом в области глубокого обучения и развитием новых методов анализа текста точность и эффективность обнаружения плагиата с помощью нейронных сетей будут повышаться.

3 Обзор приложений неповрежденных сетей

При обнаружении плагиата нейронные сети обучаются на больших объемах текстовых данных, чтобы научиться распознавать сходство между разными текстами. Эти сети могут выявлять плагиат в двух основных формах: перефразировании и пэчворке. Перефразирование предполагает переписывание текста другими словами, а пэчворк — объединение нескольких источников в один текст[5].

При анализе текста нейронные сети делят текст на более мелкие части, такие как предложения или слова, и сравнивают их с большим объемом существующего текста в базе данных. Если нейронная сеть выявляет высокую степень сходства анализируемого текста с одним из исходных текстов, соответствующая часть текста может быть помечена как потенциально плагиатная. Этот процесс может выполняться быстро и автоматически, что позволяет эффективно сканировать большое количество текстов и выявлять случаи плагиата[6].

Нейронные сети, особенно глубокое обучение и рекуррентные нейронные сети (RNN), доказали свою эффективность в обнаружении плагиата благодаря их способности извлекать и сравнивать строки извлечения из текстовых данных. Чтобы лучше понять, как нейронные сети работают при обнаружении плагиата, давайте обсудим некоторые ключевые концепции и методы:

- Векторное и текстовое представление – для обработки текста нейронными сетями необходимо преобразовывать слова или предложения в числовые векторы. Этот метод преобразования называется внедрением и создает векторное пространство, в котором похожие слова или фразы группируются ближе друг к другу. Общие методы встраивания включают Word2Vec, GloVe или BERT[7].
- Рекуррентные нейронные сети (RNN) — RNN специально разработаны для работы с временными рядами или последовательностями, такими как текстовые данные[8]. RNN способны фиксировать контекст и зависимости между словами или предложениями в тексте, что является ключом к анализу сходства между двумя текстами. RNN могут быть расширены слоями LSTM (Long Short-Term Memory) или GRU (Gated Recurrent Unit), которые улучшают способность сети сохранять долгосрочные данные. зависимости в последовательностях.
- Сиамские нейронные сети — этот тип нейронной сети используется для сравнения двух или более входных данных, чтобы определить, похожи они или нет. Сиамские сети обучаются на парах похожих и разных входных данных, чтобы научиться распознавать сходства и различия между текстами. Эти сети часто используются для выявления перефразирования или пэчворка в потенциально плагиатных текстах[9].
- Сверточные нейронные сети (CNN) – CNN изначально были разработаны для анализа изображений, но их также можно эффективно использовать для анализа текста. Сверточные слои в сети извлекают локальные признаки из входного текста, например, n-граммы или неполные словосочетания. Эти признаки затем используются для выявления сходства между проанализированным текстом и текстами в базе данных[10]. Сверточные нейронные сети можно комбинировать с рекуррентными нейронными сетями (RNN) или сиамскими нейронными сетями для еще более точного анализа текста.

После успешного обучения и проверки нейронную сеть можно использовать для анализа новых, немаркированных текстов. Если нейронная сеть выявляет высокую степень сходства анализируемого текста с одним из исходных текстов, соответствующая часть текста может быть помечена как потенциально плагиатная. Затем эти идентифицированные части текста можно проверить вручную, чтобы убедиться, что это действительно плагиат.

4 Кейсы и исследования

Чтобы еще глубже погрузиться в работу NN в мире плагиата, мы построили действующий пример нейронной сети. Наша нейронная сеть рассмотрит два вопроса и попытается определить, имеют ли два вопроса одинаковое значение (это та же концепция, что и (большинство современных детекторов плагиата сравнивают целые предложения или иногда абзацы текста, чтобы увидеть, идентичны ли они семантически). Для этой цели мы выбрали модель RNN, которая чрезвычайно подходит для этой задачи благодаря своей способности обрабатывать последовательные данные, такие как текст.
Мы использовали вариант LSTM, потому что он лучше приспособлен для обработки долгосрочных зависимостей в тексте. В нашем случае (сравнение двух предложений) важная информация может быть найдена далеко друг от друга в контексте одного предложения, что было бы сложной задачей для обычных RNN.

Все начинается с обработки данных. На входе у нас есть набор данных, содержащий 10 000 парных вопросов и показатель значимости (одинаковы ли пары по смыслу или нет). Предложения предварительно обрабатываются (текст в предложениях пишется строчными буквами, а затем предложения разбиваются на список слов (токенов)). Комбинации двух предложений (в виде обработанных списков) вставляются в модель Word2Vec (модель Word2Vec включает изучение вложенных слов, которые могут сохранять семантические отношения между словами на основе их общего употребления в корпусе текстов).
Word2Vec позволяет узнать, насколько сильны связи между словами (коэффициенты корреляции). Skip-gram используется для обнаружения вложенных слов в контексте Word2Vec: по результатам исследования [13] он имеет наилучшие параметры. Кроме того, необходимо уравнять количество слов в предложениях, т.е. добавить «пустые места» (нулевые значения) в конец меньшего предложения. Это сделано для того, чтобы матрицы, которые перемножаются в NS, имели одинаковую длину.

Мы использовали 80% нашего набора данных в качестве данных для обучения и 20% для тестирования и оценки. После обработки всех данных наша модель LSTM была обучена. LSTMModel — это однослойная сеть LSTM, за которой следует линейный слой. Слой LSTM имеет размер входных данных 128 (размер входных данных Word2Vec) и размер скрытого слоя 128. Скорость обучения установлена на 0,001, количество эпох равно 15, а размер пакета — 32 единицы. В качестве оптимизатора мы выбрали Адама (несмотря на состояние, оно непротиворечиво). Функция активации, используемая в этом случае, является сигмовидной, так что в конце мы имеем небольшие вероятности. Мы выбрали бинарную перекрестную энтропию в качестве функции потерь, потому что у нас есть только два возможных выхода (одинаковые предложения или разные). Разница между результатами для двух вопросов указывает на вероятность дублирования вопросов. Благодаря обратному распространению мы можем эффективно регулировать веса в нашей модели.

LSTM-модель

```
класс LSTMModel (nn.Module):
    def __init__(self, input_size, hidden_size, num_layers): super(LSTMModel,
        self).__init__() self.lstm = nn.LSTM(input_size,
        hidden_size, num_layers, batch_first=True) self.fc = nn.Linear(hidden_size, 1)


    def forward(self, x): _,
        (hidden, _) = self.lstm(x) output =
        self.fc(hidden[-1]) return
        torch.sigmoid(output)
```

Обработка данных

```
url = "https://raw.githubusercontent.com/S4IKEz/stuff/main/questions1.csv" data = pd.read_csv(url,
usecols=['question1', 'question2', 'is_duplicate'])

данные['вопрос1'] = данные['вопрос1'].str.lower().str.split() данные['вопрос2'] =
данные['вопрос2'].str.lower().str.split()

# Обучение модели Word2Vec
Offerings = data['question1'].tolist() + data['question2'].tolist() model_w2v =
Word2Vec(sentences, vector_size=128, window=5, min_count=1, workers=4 )

# Кодировать
Word2Vec def encode_questions(вопрос):
    return np.array([model_w2v.wv[word] для рассматриваемого слова])

данные['q1_encoded'] = данные['question1'].apply(encode_questions)
данные['q2_encoded'] = данные['question2'].apply(encode_questions)

X1 = pad_sequences(данные['q1_encoded'].tolist())
X2 = pad_sequences(data['q2_encoded'].tolist()) y =
data['is_duplicate'].values

X1_train, X1_test, X2_train, X2_test, y_train, y_test = train_test_split(X1, X2, y,
test_size=0,2, random_state=42)
```

Параметры

```
input_size = 128
hidden_size = 128
num_layers = 1
num_epochs = 15
Learning_rate = 0,001 batch_size
= 32
```

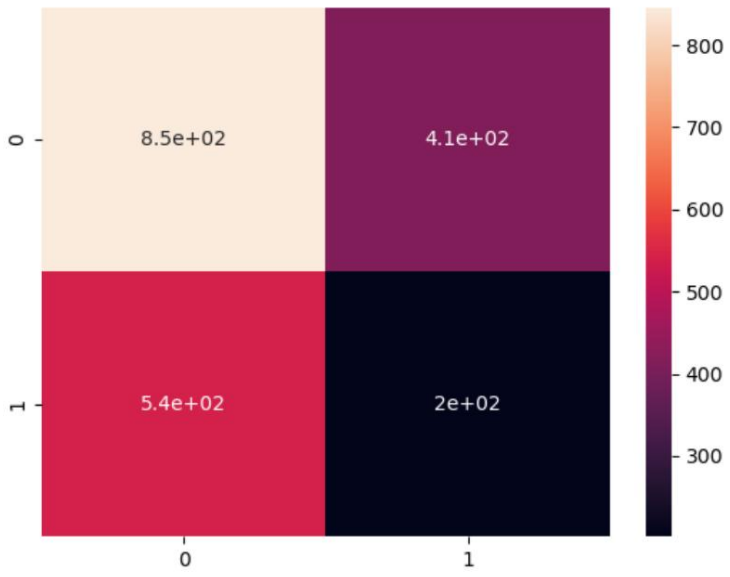
Обучение

```
model = LSTMModel (input_size, hidden_size, num_layers) критерий = nn.BCELoss()
оптимизатор = optim.Adam
(model.parameters(), lr = learning_rate)

model.train() для
эпохи в диапазоне (num_epochs): для i, (q1,
    q2, метки) в enumerate(train_loader): q1_out = model(q1) q2_out = model(q2)
        # print(q1_out, q2_out);
        выход = факел.abs (q1_out
        - q2_out)


        потеря = критерий (аут, метки)

        оптимизатор.ноль_град()
        потеря.назад()
        оптимизатор.шаг()
```

Результаты

```
model.eval() c
torch.no_grad(): X1_test_tensor
    = torch.tensor(X1_test, dtype=torch.float32)
    X2_test_tensor = torch.tensor(X2_test, dtype=torch.float32) q1_test_out =
    model(X1_test_tensor) q2_test_out =
    model(X2_test_tensor) test_out = torch.abs(q1_test_out
    - q2_test_out) test_preds = (test_out > 0.5).type(torch.float32) .view(-1)
    y_true = torch.tensor(y_test, dtype=torch.float32).view(-1) y_pred = test_preds cm =
    путаница_матрица(y_true, y_pred) print("Матрица путаницы:") print(cm) test_acc =
    torch.mean((test_preds ==
    torch.tensor(y_test,


    dtype=torch.float32)) .type(torch.float32)).item() print(f"Точность: {test_acc}")
```

После обучения модели мы протестируем ее и проверим производительность нашей NN, используя матрицу путаницы. Общий результат будет процентом правильно классифицированных образцов.



инжир. 1: Матрица путаницы

5 Обсуждение

В этом разделе мы рассмотрим результаты вышеупомянутого исследования и более подробно рассмотрим ключевые критерии сравнения. Во-первых, применение векторного представления к тексту преобразует текст в список векторов, которые сохраняют семантические и синтаксические свойства, предоставляемые алгоритмом глубокого обучения. Во-вторых, критерий уровня обработки определяет, обрабатывается ли текст на уровне слов или предложений. Наконец, метод подобия относится к подходам, используемым для вычисления сходства между векторами, представляющими тексты, и обеспечивает обзор сильных и слабых страниц каждого «моего метода».

В большинстве подходов используется метод Word2Vec или Doc2Vec для векторного преобразования, а наиболее эффективным методом сохранения смысловой стороны заданного текста является представление Миколовского. Преобразование текста в список предложений является наиболее подходящим представлением, поскольку оно учитывает смысл текста.

Относительно методов, используемых для расчета сходства: используются разные подходы для определения наличия сходства между анализируемыми текстами. Многие из этих подходов используют в своей архитектуре CNN и RNN, но большинство из них полагаются на векторное представление на уровне слов. Это ограничение снижает их способность обнаруживать сходство между целыми предложениями или текстами, вместо этого сосредотачиваясь только на словах.

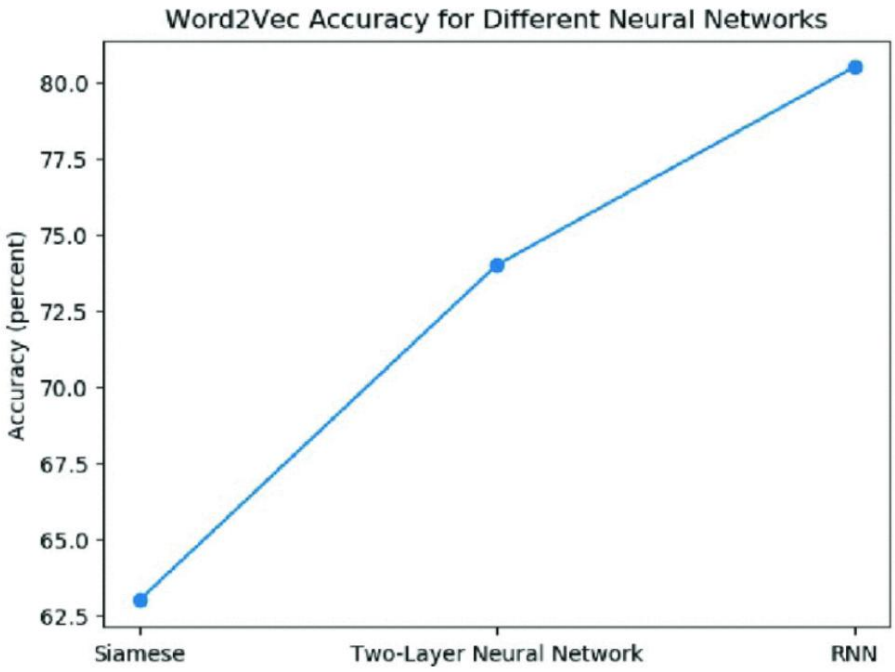
Почти все подходы используют косинус для вычисления сходства между документами, что делается пословно или по предложениям. Однако этот метод может привести к ненадежным результатам, поскольку два документа могут содержать одно и то же слово или предложение, не будучи семантически идентичными. Для решения этой проблемы необходим подход, который представляет текст как список предложений, который преобразуется в список векторов. Обработка, применяемая к списку предложений, должна сохранять семантический аспект текста, и для обнаружения сходства можно использовать такой алгоритм, как RNN.[12]

Еще одним преимуществом является то, что нейронные сети можно использовать в различных отраслях и контекстах. Их можно использовать в академических кругах для проверки студенческих работ на плагиат, в издательском бизнесе для проверки дублированного контента и даже в юридической сфере для выявления случаев нарушения авторских прав.

Доступно несколько инструментов и платформ для использования нейронных сетей при обнаружении плагиата. Среди самых популярных — Turnitin, iThenticate и PlagScan. Эти платформы позволяют пользователям загружать документы и анализировать их на наличие потенциальных случаев плагиата.

6 Оценка аз'авер

Плагиат становится все более распространенной проблемой, особенно с появлением таких технологий, как ChatGPT и других. Неприятное чувство, когда кто-то пытается украсть твою идею или незаконно ее скопировать. Поэтому мы решили написать эту статью, чтобы подчеркнуть важность использования нейронных сетей в борьбе с плагиатом. Используя RNN и метод Word2Vec, мы добились точности обнаружения плагиата в 61%, что, на наш взгляд, является достойным результатом.
Ниже представлен график «Точность метода Word2Vec для разных НС»:



инжир. 2: Точность Word2Vec для различных нелинейных сетей[13]

В заключение можно сказать, что использование нейронных сетей при выявлении плагиата является эффективным и действенным способом борьбы с несанкционированным копированием в различных домах. С ростом доступности цифровых данных использование нейронных сетей становится все более популярным во многих отраслях. Поэтому крайне важно, чтобы отдельные лица и организации ознакомились с этой технологией и внедрили ее в свою деятельность, чтобы гарантировать целостность и подлинность своего контента.

Список использованной литературы

[1] Джоннес, Дж. (2004). Империи света: Эдисон, Тесла, Вестингауз и гонка за электрификацию мира. Рэндом Хаус Торговля в мягкой обложке.

[2] Али, А., и Така, А.Я. (2022). Аналитическое исследование традиционных и интеллектуальных подходов к обнаружению текстового плагиата. Журнал образования и науки, 31 (1), 8-25.

[3] Девлин, Дж., Чанг, М.В., Ли, К., и Тутанова, К. (2018). BERT: предварительная подготовка глубоких двунаправленных преобразователей для понимания языка. Препринт arXiv arXiv: 1810.04805.

[4] Эль-Рашиди, Массачусетс, Мохамед, Р.Г., Эль-Фишави, Н.А., и Шуман, Массачусетс (2022). Надежная система обнаружения плагиата, основанная на подходах глубокого обучения. Нейронные вычисления и приложения, 34 (21), 18837-18858.

[5] Гарави, Э., Вейси, Х., и Россо, П. (2020). Масштабируемый и независимый от языка подход к обнаружению плагиата, основанный на внедрении, с учетом типа обфускации: без этапа обучения. Нейронные вычисления и приложения, 32, 10593-10607.

[6] Альзахрани, С.М., Салим, Н., и Абрахам, А. (2012). Понимание языковых моделей плагиата, текстовых особенностей и методов обнаружения. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42, 133-149.

[7] Минаи, С., Калхбреннер, Н., Камбрия, Э., Никзад, Н., Ченаглу, М., и Гао, Дж. (2021). Классификация текстов на основе глубокого обучения: всесторонний обзор. Вычислительные исследования ACM (CSUR), 54(3), 1-40.

[8] Кулкарни, С., Говилкар, С., и Амин, Д. (2021, май). Анализ средств и методов обнаружения плагиата. В материалах 4-й Международной конференции по достижениям в области науки и технологий (ICAST2021).

[9] Тянь З., Ван К., Гао К., Чен Л. и Ву Д. (2020). Обнаружение плагиата многопоточных программ с помощью сиамских нейронных сетей. Доступ IEEE, 8, 160802-160814.

[10] Бенаббоу, Ф. (2020). Новая онлайн-система обнаружения плагиата, основанная на глубоком обучении. Международный журнал передовых компьютерных наук и приложений, 11 (9).

[11] Шаоцзе Бай, Дж. Зико Колтер, Владлен Колтун. Эмпирическая оценка общих сверточных и рекуррентных сетей для моделирования последовательностей. arXiv:1803.01271v2 [cs.LG] 19 апреля 2018 г.

[12] Куок Ле и Томас Миколов. Распределенные представления предложений и документов. Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043.

[13] Хант, Э., Джанамсетти, Р., Кинарес, К., Кох, К., Санчес, А., Жан, Ф., ... и О, П. (2019, ноябрь). Модели машинного обучения для идентификации парафраз и их применения для обнаружения плагиата. В 2019 году Международная конференция IEEE по большим знаниям (ICBK) (стр. 97-104). IEEE.