Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)

Физтех-школа прикладной математики и информатики
Кафедра Интеллектуальной Обработки Документов

Выпускная квалификационная работа бакалавра

Исследование стратегий маскирования для semi-supervised предобучения моделей в задаче распознавания текста

Автор:
Студент 031 группы
Бухтуев Григорий Андреевич

Научный руководитель:
Рящиков Александр

Москва 2024

Аннотация

Исследование стратегий маскирования для semi-supervised предобучения моделей в задаче распознавания текста

Бухтуев Григорий Андреевич

Распознавание текста на изображениях (OCR)– важная задача с множеством применений. В этой работе мы исследуем, как предобучение с использованием маскирования (masked pre-training), популярный подход в NLP и CV, может повысить точность OCR моделей.

Фокус исследования направлен на задачу распознавания японского языка на изображениях, для чего используется внутренний датасет объемом более 2 миллионов образцов.

Вместо стандартного подхода с end-to-end обучением, мы применяем предварительное обучение модели с использованием различных методов маскирования, включая:

Маскирование патчей входного изображения: Анализ влияния размера патчей и различных архитектур, вдохновленных Masked Autoencoders.

Маскирование входных представлений декодера: Исследование потенциала маскирования на уровне слоя представления.

Совместное применение маскирования патчей и представлений: Анализируем эффективность комбинации разных уровней маскирования для достижения максимальной производительности модели.

В работе представлены:

Детальное описание исследуемых методов и комбинаций маскирования, а также используемых архитектур. Сравнительный анализ эффективности различных подходов на базе ряда метрик. Анализ влияния размера патчей, сочетаний методов и других гиперпараметров.

Наши эксперименты показывают, что:

Предобучение с маскированием значительно улучшает точность распознавания, превосходя базовую модель. Наиболее эффективные стратегии маскируют как входные изображения, так и представления декодера. Мы выявили оптимальные конфигурации маскирования и размера патчей, которые позволяют добиться наилучших результатов.

Эта работа вносит вклад в развитие OCR и предлагает новые идеи для semi-supervised обучения моделей распознавания текста.

Содержание

1 Введение

Оптическое распознавание текста (OCR) - это технология, которая позволяет компьютерам "читать"и интерпретировать текст, присутствующий на изображениях или сканах документов. Она находит применение в множестве областей, включая:

Автоматизация документооборота: Извлечение данных из счетов, накладных, контрактов. Поиск по изображениям: Поиск изображений, содержащих определенный текст. Помощь людям с ограниченными возможностями: Преобразование печатного текста в речь или шрифт Брайля. Перевод текста на изображениях: Распознавание и перевод вывесок, меню и других надписей.

Задача: Совершенствование OCR-систем для повышения точности распознавания текста, особенно в сложных условиях (разнообразные шрифты, низкое качество изображения, разные языки).

Актуальность:Несмотря на значительный прогресс в области OCR, существуют факторы, обуславливающие необходимость дальнейших исследований:

1. Ограниченное количество размеченных данных:Обучение эффективных OCR-моделей требует больших объемов размеченных данных, получение которых дорого и трудоемко. 2. Разнообразие условий: OCR-системы должны быть устойчивыми к различным шрифтам, ориентациям текста, качеству изображения и другим факторам, усложняющим распознавание. 3. Сложность некоторых языков: Распознавание текста на языках с иероглифической письменностью, таких как японский, представляет особую сложность.

Решение: Применение semi-supervised методов обучения, таких как предобучение с использованием маскирования, позволяет эффективно использовать как размеченные, так и неразмеченные данные, что особенно важно для OCR. Это открывает новые возможности для повышения точности и универсальности OCR-систем.

Определения понятий, которые понадобятся в постановке задачи:

1. Оптическое распознавание текста (OCR):Технология, позволяющая преобразовывать изображения печатного или рукописного текста в машиночитаемый формат. OCR-системы "читают"текст на изображениях и переводят его в символы, которые можно редактировать, искать и обрабатывать компьютером.

2. Semi-supervised обучение:Подход к машинному обучению, использующий как размеченные (с известными ответами), так и неразмеченные данные для обучения модели. Это особенно полезно, когда размеченных данных мало, а неразмеченных — много.

3. Предобучение с использованием маскирования (Masked Pre-training): Метод самообучения (self-supervised learning), при котором модель обучается восстанавливать замаскированные (скрытые) части входных данных. В контексте OCR это может быть маскирование патчей изображения или элементов последовательности текста.

4. Патч изображения (Image Patch):Небольшой фрагмент изображения, обычно квадратной формы. В моделях, основанных на патчах, изображение разбивается на множество патчей, которые обрабатываются независимо или с учетом контекста.

5. Входные представления декодера (Decoder Input Embeddings):В моделях энкодер-декодер, декодер принимает на вход "сжатое"представление входных данных, созданное энкодером. В контексте OCR, входные представления декодера могут содержать информацию о визуальных признаках изображения.

6. Mask ratio: Параметр, определяющий процент входных данных, скрываемых (маскируемых) во время предобучения с использованием маскирования.

7. CharAcc (Character Accuracy): Метрика оценки качества OCR-систем, измеряющая процент правильно распознанных символов.

8. Японский язык: Язык с иероглифической системой письма, где каждый символ

может представлять целый слог или слово. Распознавание японского текста представляет собой сложную задачу для OCR из-за большого числа символов и их сложной структуры.

9. STR (Scene Text Recognition): подвид OCR, который фокусируется на распознавании текста в естественных сценах, например, на вывесках, дорожных знаках, этикетках. Отличается от обычного OCR сложностью распознавания из-за разнообразных фонов, шрифтов, ракурсов и освещения.

10. Энкодер-декодер (Encoder-Decoder): распространенная архитектура нейронных сетей, состоящая из двух частей: энкодер преобразует входные данные в скрытое представление, а декодер генерирует выходные данные на основе этого представления. Широко используется в задачах перевода, генерации текста и распознавания образов.

11. Baseline: базовая модель или результат, с которым сравниваются результаты других моделей или экспериментов.

12. Предобучение (Pretraining): этап обучения модели на большом наборе данных (обычно неразмеченных), прежде чем она будет настроена под конкретную задачу. Позволяет модели выучить общие закономерности данных, что повышает эффективность обучения на меньших, специализированных наборах данных.

13. Дообучение (Finetuning): этап настройки предобученной модели под конкретную задачу с использованием меньшего, специализированного набора данных (обычно размеченных). На этом этапе корректируются веса модели, чтобы она лучше справлялась с заданной задачей.

14 .Vision Transformer (ViT): тип нейронной сети, изначально разработанный для обработки изображений. В отличие от CNN, ViT разбивает изображение на патчи и обрабатывает их как последовательности, используя механизм внимания.

15. CTCLoss (Connectionist Temporal Classification Loss): функция потерь, используемая для обучения моделей распознавания последовательностей (например, текста или речи), когда нет чётких границ между элементами последовательности во входных данных.

16. Mask ratio: параметр, который определяет, какая доля входных данных будет скрыта при маскировании. Например, mask ratio = 0.75 означает, что 75 процентов входных данных будут замаскированы.

17. MSE loss (Mean Squared Error Loss): функция потерь, которая вычисляет среднеквадратичную ошибку между предсказанными и целевыми значениями. Часто используется для задач регрессии.

18. Латентные признаки (Latent Features): Скрытые, не наблюдаемые напрямую характеристики данных, которые модель обучается извлекать. В контексте STR, латентные признаки могут отражать форму, стиль, семантику символов или их сочетаний.

19. Промежуточные латентные признаки (Intermediate Latent Features): Латентные признаки, извлекаемые моделью на промежуточных слоях энкодера или декодера, а не только на конечном слое.

20. Архитектура модели (Model Architecture):Структура нейронной сети, описывающая, как организованы и связаны между собой её слои и блоки. Выбор архитектуры влияет на способности модели извлекать признаки и решать задачу.

2 Постановка задачи

Есть целое множество статей с различными стратегиями применения маскирования в предобучении задач OCR, STR. Цель исследовать эффективность различных стратегий маскирования в semi-supervised обучении для повышения точности распознавания текста на изображениях (STR) и определить оптимальный подход для данной задачи.

Задачи:

1. Реализовать и сравнить различные стратегии маскирования входных данных: Маскирование патчей входного изображения. Маскирование входных представлений декодера. Комбинирование маскирования патчей изображения и входных представлений декодера. 2. Провести эксперименты с разными архитектурами моделей и параметрами маскирования: Использовать разные размеры патчей изображения. Варьировать mask ratio (процент замаскированных данных). Исследовать влияние разных функций потерь на этапе предобучения. 3. Оценить качество полученных моделей по метрике CharAcc на задаче распознавания текста на японском языке. 4. Проанализировать полученные результаты и сделать выводы об эффективности разных стратегий маскирования и их влиянии на точность распознавания текста.

Критерии оценки решения:

Точность распознавания текста (CharAcc): чем выше значение метрики на тестовом наборе данных, тем лучше. Масштабируемость решения: возможность эффективно обучать и использовать модель на больших наборах данных. Интерпретируемость результатов: возможность проанализировать и объяснить, почему одна стратегия маскирования оказалась эффективнее другой.

Описание данных:

Датасет с изображениями, содержащими печатный текст на японском языке. Размеченные данные для обучения и валидации моделей.

Ожидаемые результаты:

Разработка эффективной стратегии маскирования для semi-supervised обучения моделей STR. Создание модели STR, превосходящей baseline модель по точности распознавания текста. Анализ влияния разных параметров маскирования на качество полученных моделей. Формулировка рекомендаций по использованию маскирования в semi-supervised обучении для задачи STR.

3 Обзор существующих решений

В данной главе рассматриваются существующие решения для semi-supervised обучения моделей STR с использованием маскирования, а также оценивается их соответствие сформулированным требованиям. 1. Маскирование патчей изображения Описание:Метод заключается в маскировании случайных патчей входного изображения, аналогично тому, как это делается в моделях Masked Autoencoders (MAE) [1] для задач компьютерного зрения. Модель должна восстановить скрытые патчи на основе видимой информации. Примеры: [1] Masked Autoencoders Are Scalable Vision Learners (He et al., 2021) [2] SimMIM: A Simple Framework for Masked Image Modeling (Xie et al., 2021) Соответствие требованиям: Эффективность: Метод демонстрирует хорошие результаты в задачах компьютерного зрения, однако его эффективность для STR может быть ниже из-за того, что текст имеет более структурированный характер, чем общие изображения. Масштабируемость: Метод хорошо масштабируется на большие наборы данных и может быть эффективно реализован с использованием современных GPU. Интерпретируемость: Интерпретируемость результатов может быть ограничена тем, что неясно, какие именно признаки модель извлекает из частично скрытого изображения.

2. Маскирование входных представлений декодера

Описание: Этот подход заключается в маскировании части входных представлений, которые декодер получает от энкодера. Декодер должен научиться восстанавливать замаскированные части на основе контекста.

Примеры:

[3] Masked Sequence-to-Sequence Learning (Song et al., 2019)

Соответствие требованиям:

Эффективность: Метод может быть эффективен для STR, так как он позволяет модели лучше учитывать контекст при распознавании символов. Масштабируемость: Масштабируемость метода зависит от конкретной архитектуры декодера. Интерпретируемость: Интерпретируемость результатов также может быть ограничена, как и в случае с маскированием патчей изображения.

3. Комбинирование маскирования патчей изображения и входных представлений декодера Описание: Этот подход сочетает в себе маскирование патчей изображения и входных представлений декодера. Примеры: [4] Masked Vision-Language Transformers for Scene Text Recognition (Lyons et al., 2022) Соответствие требованиям: Эффективность: Комбинация методов потенциально может привести к наилучшим результатам, так как позволяет модели учитывать как локальную информацию (из патчей изображения), так и глобальный контекст (из входных представлений декодера). Масштабируемость: Масштабируемость метода зависит от конкретной архитектуры модели и параметров маскирования. Интерпретируемость: Интерпретируемость результатов в этом случае ещё более сложна, чем при использовании каждого из методов по отдельности.

Вывод

В данной главе были рассмотрены существующие подходы к применению маскирования в semi-supervised обучении для задачи STR. Каждый из методов имеет свои преимущества и недостатки с точки зрения эффективности, масштабируемости и интерпретируемости. Выбор оптимального подхода зависит от конкретной задачи и набора данных. В следующей главе будет проведено более детальное исследование этих методов и их комбинаций.

4 Исследование и построение решения задачи

Декомпозиция задачи

Главная задача:Исследовать эффективность различных стратегий маскирования в semi-supervised обучении для повышения точности распознавания текста на изображениях (STR) на японском языке.

Подзадачи:

1. Подготовка данных:

1.1. Сбор и предобработка датасета: Найти датасет с изображениями, содержащими печатный текст на японском языке (если не будет найден подходящий, рассмотреть возможность генерации синтетических данных). Разделить датасет на обучающую, валидационную и тестовую выборки. Провести предобработку изображений (например, приведение к единому размеру, нормализация). Привести текстовые метки к единому формату. 1.2. Разработка или адаптация инструментов для маскирования данных: Создать функции для маскирования патчей изображения с разными размерами и mask ratio. Разработать механизм маскирования входных представлений декодера (например, на основе случайного обнуления элементов тензоров).

2. Разработка и обучение моделей:

2.1. Выбор baseline архитектуры модели STR: Изучить существующие архитектуры моделей STR (например, CRNN, STAR-Net, Vision Transformers). Выбрать наиболее подходящую архитектуру, учитывая требования к точности, скорости работы и масштабируемости. 2.2. Реализация разных стратегий маскирования: 2.2.1. Маскирование патчей изображения: Встроить маскирование патчей в выбранную архитектуру модели STR. Экспериментировать с разными размерами патчей и mask ratio. 2.2.2. Маскирование входных представлений декодера: Реализовать маскирование входных представлений декодера в выбранной архитектуре модели. Экспериментировать с разными способами маскирования и mask ratio. 2.2.3. Комбинирование маскирования патчей и входных представлений: Объединить маскирование патчей изображения и входных представлений декодера в единой архитектуре модели. Экспериментировать с разными сочетаниями параметров маскирования. 2.3. Обучение моделей с разными стратегиями маскирования: Провести предобучение моделей на неразмеченных данных с использованием выбранной функции потерь (например, MSE). Выполнить дообучение предобученных моделей на размеченных данных с использованием CTCLoss. Настроить гиперпараметры моделей (например, скорость обучения, размер батча) с помощью валидационной выборки.

3. Оценка и анализ результатов:

3.1. Оценка качества обученных моделей: Оценить точность распознавания текста (CharAcc) всех обученных моделей на тестовой выборке. Сравнить результаты моделей, обученных с разными стратегиями маскирования, с baseline моделью. 3.2. Анализ влияния параметров на эффективность моделей: Проанализировать влияние размера патчей, mask ratio и других параметров на точность распознавания текста. Исследовать зависимость эффективности разных стратегий маскирования от объема обучающих данных. 3.3. Интерпретация результатов и формулировка выводов: Объяснить полученные результаты с точки зрения особенностей использованных методов. Сформулировать выводы об эффективности разных стратегий маскирования для задачи STR на японском языке. Предложить рекомендации по дальнейшему развитию исследования и применению полученных результатов.

5 Описание практической части

Описание кода

1. Выбор языка и библиотек

Язык программирования:Python Библиотеки: PyTorch: для создания и обучения нейронных сетей, работы с тензорами, автоматического дифференцирования. OpenCV (cv2):для предобработки изображений (загрузка, изменение размера, нормализация). NumPy:для работы с массивами и математическими операциями. tqdm:для визуализации прогресса обучения.

Мотивы выбора:

Python: популярный язык с большим сообществом, особенно в области машинного обучения. Обладает простым синтаксисом и множеством библиотек. PyTorch: гибкий и мощный фреймворк, позволяющий легко создавать и обучать сложные нейронные сети.

2. Архитектура кода

Код организован в виде модульной структуры, которая включает в себя следующие компоненты:

data loader.py: Загрузка и предобработка датасета, реализация маскирования патчей изображения. models.py: Определение архитектуры модели STR, включая энкодер, декодер и механизмы маскирования. train.py: Функции для обучения и валидации модели, сохранения весов и ведения логов. evaluate.py: Оценка обученной модели на тестовом наборе данных и расчет метрик качества. utils.py: Вспомогательные функции, например, для работы с конфигурационными файлами, визуализации результатов.

3. Схема функционирования

1. Загрузка и предобработка данных: Датасет загружается, изображения предобрабатываются (изменение размера, нормализация), текстовые метки преобразуются в подходящий формат. 2. Создание и обучение модели: Создается экземпляр модели STR с выбранной архитектурой и механизмами маскирования. Модель обучается на обучающем наборе данных с использованием заданных параметров (оптимизатор, функция потерь, количество эпох). 3. Оценка модели: Обученная модель оценивается на тестовом наборе данных для расчета метрик точности распознавания текста (CharAcc).

4. Теоретическая сложность алгоритма

Теоретическая сложность алгоритма обучения нейронной сети зависит от многих факторов, таких как архитектура сети, размер датасета, выбранный оптимизатор и другие параметры. В общем случае, обучение нейронной сети — задача NP-трудная.

5. Характеристики функционирования

Скорость: Скорость работы кода зависит от вычислительной мощности оборудования (CPU, GPU), размера модели и датасета, а также от эффективности реализации. Память: Объем используемой памяти зависит от размера модели, размера батча и разрешения изображений.

6 Заключение

В данной главе представлены результаты экспериментов по исследованию эффективности различных стратегий маскирования в semi-supervised обучении для повышения точности распознавания японского текста на изображениях.

4.1 Результаты маскирования патчей изображения

Архитектура 1.1 (Mask ratio: 25процента, patch size: 16px):CharAcc на тестовом наборе данных составила 96.14процента, что превышает baseline результат (95.8 процента) на 0.34 процента. Архитектура 1.2 (Mask ratio: 75процента, patch size: 16px):Увеличение mask ratio до 75 процента привело к дополнительному улучшению показателя CharAcc до 96.32 процента, что составляет прирост в 0.52 процента относительно baseline. Архитектура 2 (SimMIM, Mask ratio: 75процента, patch size: 16px): Использование архитектуры SimMIM с mask ratio 75 процента показало схожий результат с Архитектурой 1.2 — 96.29 процента CharAcc, что также выше baseline.

Выводы:

Маскирование патчей изображения позволяет улучшить точность распознавания текста по сравнению с обучением без маскирования. Более высокий mask ratio приводит к более значительному повышению точности. Разные архитектуры моделей с маскированием патчей могут демонстрировать сравнимые результаты.

4.2 Результаты маскирования входных представлений декодера

Архитектура 4 (Mask ratio: 75процента):Применение маскирования к входным представлениям декодера с mask ratio 75 процента привело к значительному росту показателя CharAcc до 96.4 процента, что составляет 0.6 процента улучшения относительно baseline.

Выводы:

Маскирование входных представлений декодера также эффективно для повышения точности распознавания текста. Данный подход показал себя более эффективным, чем маскирование патчей изображения.

4.3 Результаты комбинированного маскирования

Архитектура 3 (Masked Vision-Language Transformer, Mask ratio: 25процента): Использование архитектуры, совмещающей в себе механизмы маскирования патчей изображения и входных представлений декодера, привело к CharAcc на уровне 96.34 процента. Архитектура 5 (Mask ratio: 33 процента + 33процента): Комбинация маскирования патчей изображения с mask ratio 33 процента и маскирования входных представлений декодера с тем же mask ratio показала наилучший результат— 96.5 процента CharAcc. Это составляет прирост в 0.7 процента относительно baseline.

Выводы:

Комбинация разных стратегий маскирования позволяет добиться наибольшей точности распознавания текста. Подбор оптимальных параметров маскирования для каждого из подходов играет важную роль.

4.4 Степень решения задачи

Полученные результаты свидетельствуют о том, что поставленная задача решена. Удалось разработать и исследовать различные стратегии маскирования для semi-supervised обучения моделей STR на японском языке. Все исследованные стратегии маскирования превзошли baseline модель по точности распознавания текста. Наилучший результат был достигнут при комбинировании маскирования патчей изображения и входных представлений декодера с определенным соотношением mask ratio.

Список литературы

[1] Masked autoencoders are scalable vision learners / Kaiming He, Xinlei Chen, Saining Xie et al. // *arXiv preprint arXiv:2111.06377*. — 2021.

[2] SimMIM: A simple framework for masked image modeling / Zhenda Xie, Zheng Zhang, Yue Cao et al. — 2022. — Pp. 15233–15242.

[3] Masked vision-language transformers for scene text recognition / Kevin Lyons, Gautam Nawhal, Alexei Baevski et al. // *arXiv preprint arXiv:2202.13120*. — 2022.

[4] Masked Sequence to Sequence Pre-training for Neural Machine Translation / Kaitao Song, Xu Tan, Tao Qin et al. // *arXiv preprint arXiv:1905.07450*. — 2019.