

IPDF2LATEX

$$\det \left(\frac{\partial^2 F}{\partial x_i \partial x_j} \right) = \sum_{\sigma \in S_n} \text{sgn}(\sigma)$$

`\det\left(\frac{\partial^2 F}{\partial x_i \partial x_j} \right) = \sum_{\sigma \in S_n} \text{sgn}(\sigma)`

Macéo Ottavy, Mathieu Longatte, Louison Mocq, Ankit Gayen

Supervised by Simon Delamare

and the rule is proved that

$$\frac{du^n}{dx} = nu^{n-1} \frac{du}{dx},$$

where n is a positive fraction whose numerator and denominator are integers. This rule has already been used in the solution of numerous exercises.

34. The Derivative of a Constant. Let $y = c$, where c is a constant. Corresponding to any Δx , $\Delta y = 0$, and consequently

$$\frac{\Delta y}{\Delta x} = 0,$$

and

$$\lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = 0,$$

or

$$\frac{dy}{dx} = 0.$$

The derivative of a constant is zero.

Interpret this result geometrically.

35. The Derivative of the Sum of Two Functions. Let

$$y = u + v,$$

where u and v are functions of x . Let Δu , Δv , and Δy be the increments of u , v , and y , respectively, corresponding to the increment Δx .

$$y + \Delta y = u + \Delta u + v + \Delta v$$

$$\Delta y = \Delta u + \Delta v$$

$$\frac{\Delta y}{\Delta x} = \frac{\Delta u}{\Delta x} + \frac{\Delta v}{\Delta x}$$

$$\frac{dy}{dx} = \frac{du}{dx} + \frac{dv}{dx},$$

or

$$\frac{d(u+v)}{dx} = \frac{du}{dx} + \frac{dv}{dx}.$$

The derivative of the sum of two functions is equal to the sum of their derivatives.

and the rule is proved that

$$\frac{du^n}{dx} = nu^{n-1} \frac{du}{dx},$$

where n is a positive fraction whose numerator and denominator are integers. This rule has already been used in the solution of numerous exercises.

34 The Derivative of a Constant

Let $y = c$, where c is a constant. Corresponding to any Δx , $\Delta y = 0$, and consequently

$$\frac{\Delta y}{\Delta x} = 0,$$

and

$$\lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = 0,$$

or

$$\frac{dy}{dx} = 0.$$

The derivative of a constant is zero.

Interpret this result geometrically.

35 The Derivative of the Sum of Two Functions

Let

$$y = u + v,$$

where u and v are functions of x . Let Δu , Δv , and Δy be the increments of u , v , and y , respectively, corresponding to the increment Δx .

$$y + \Delta y = u + \Delta u + v + \Delta v$$

$$\Delta y = \Delta u + \Delta v$$

$$\frac{\Delta y}{\Delta x} = \frac{\Delta u}{\Delta x} + \frac{\Delta v}{\Delta x}$$

$$\frac{dy}{dx} = \frac{du}{dx} + \frac{dv}{dx},$$

or

$$\frac{d(u+v)}{dx} = \frac{du}{dx} + \frac{dv}{dx}.$$

The derivative of the sum of two functions is equal to the sum of their derivatives.



Tensors

Exercises on Tensors

Solutions

SKLEARN PACKAGE

Logistic regression model

TRANSFORMERS

Quick tour

Preprocess

Summary of the tasks

Summary of the tokenizers

Fine-tune a pretrained model

CREATING A DATABASE

Creating image to latex code dataset

Generating latex code

Compile .tex document to pdf

Computing cleaner image

Creating dataset

FINETUNING A PRE-EXISTING MODEL

Introduction

Set up the environment

Set up the model

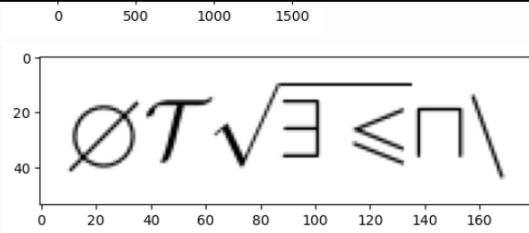
Load and preprocess the dataset

Set up the training environment

Train and evaluate the model

Results and conclusion

References



<class 'list'>

Creating dataset

We finally want to create a dataset wich will be a map containing two fields.

```
[8]: ds = {"images":[], "latex-formula":[]}
```

Then we add all the images and corresponding latex-formula we want.

```
[9]: ds["images"].append(img)
     ds["latex-formula"].append(formula)
```

Finally we save our dataset thanks to the `json` package.

```
[10]: import json

      json_object = json.dumps(ds)

      ds_name = "files/image-latex-datasets.json"
      with open(ds_name, "w") as f :
          f.write(json_object)
```

To open your dataset, do this:

```
[11]: with open(ds_name, 'r') as f:
      data = json.load(f)

      print(data["latex-formula"])

[' \emptyset \mathcal{ T \sqrt{ 9 \leq } u n }']
```

latest

Website

GitHub

Tensors

Exercises on Tensors

Solutions

SKLEARN PACKAGE

Logistic regression model

TRANSFORMERS

Quick tour

Preprocess

Summary of the tasks

Summary of the tokenizers

Fine-tune a pretrained model

CREATING A DATABASE

Creating image to latex code dataset

Generating latex code

Compile .tex document to pdf

Computing cleaner image

Creating dataset

FINETUNING A PRE-EXISTING MODEL

Introduction

Set up the environment

Set up the model

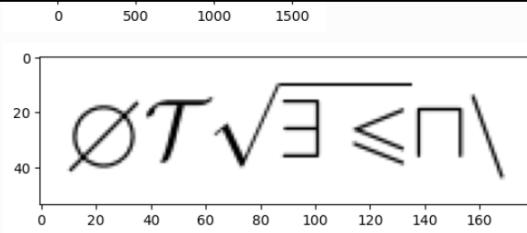
Load and preprocess the dataset

Set up the training environment

Train and evaluate the model

Results and conclusion

References



<class 'list'>

Creating dataset

We finally want to create a dataset wich will be a map containing two fields.

```
[8]: ds = {"images":[], "latex-formula":[]}
```

Then we add all the images and corresponding latex-formula we want.

```
[9]: ds["images"].append(img)
     ds["latex-formula"].append(formula)
```

Finally we save our dataset thanks to the `json` package.

```
[10]: import json

      json_object = json.dumps(ds)

      ds_name = "files/image-latex-datasets.json"
      with open(ds_name, "w") as f :
          f.write(json_object)
```

To open your dataset, do this:

```
[11]: with open(ds_name, "r") as f:
      data = json.load(f)

      print(data["latex-formula"])

[' \\emptyset \\mathcal{ T } \\sqrt{ 9 \\leq } u n ']
```

GeckSpy / IPdf2Latex

Go to file

1 Branch

0 Tags

Add file

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

IPdf2Latex

Public

Pin

Unwatch

1

Fork

Star

0

main

1 Branch

0 Tags

Go to file

Add file

Code

About

ankitgayen2021

Update metrics.ipynb

2d655c6 · 17 hours ago

191 Commits

Presentation

clean up

2 days ago

assets

update

yesterday

notebook

Update metrics.ipynb

17 hours ago

.DS_Store

changes

3 months ago

CONTRIBUTING.md

Create CONTRIBUTING.md

yesterday

README.md

update

yesterday

TODO.md

update

yesterday

tutorials-idea.md

Update tutorials-idea.md

last month

README

edit

list

IPdf2Latex

IPdf2LaTeX is a student project aimed at creating in-depth educational resources to help anyone with Python knowledge learn how to develop AI models that convert images into their LaTeX code equivalents.

We have built a website (Tutorial 1A) containing a series of notebooks that cover all the essential concepts you need.

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors

4

GeckSpy

mathieulingt

ankitgayen2021

Ankit Gayen

loumock

Objective: Image to Latex

$$\det \left(\frac{\partial^2 F}{\partial x_i \partial x_j} \right) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n \frac{\partial^2 F}{\partial x_i \partial x_{\sigma(i)}}$$



```
\det\left( \frac{\partial^2 F}{\partial x_i \partial x_j} \right)
= \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n
\frac{\partial^2 F}{\partial x_i \partial x_{\sigma(i)}}
```

Steps

Step 1: Collect data

Easy: Generate dataset from scratch or arxiv
or take one from Huggingface

Related tutorials: Basics Images Loading , Building

Steps

Step 1: Collect data

Easy: Generate dataset from scratch or arxiv
or take one from Huggingface

Related tutorials: Basics Images Loading , Building

Step 2: Chose a relevant architecture

Hard: No obvious structure for Image to text

Hard: Language models require a gigantic amount of data

Related tutorials: Curse of Dimensionality Summary of the tokenizers Fine-tune a pretrained model

Transformers

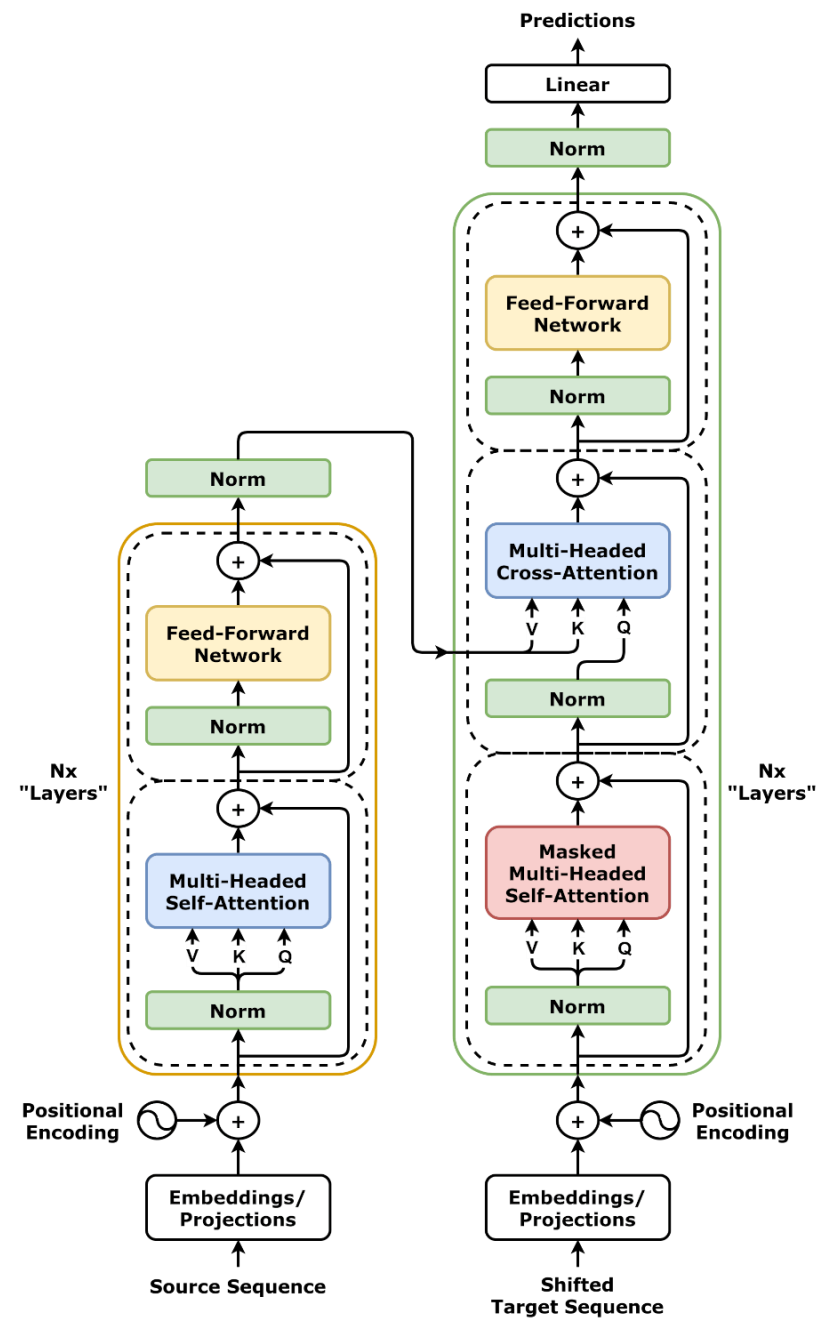
Objectives :

Capture long distance dependency

Enable parallelization in the treatment

Time :	Yesterday,
Person :	Ryan
Action :	bought
Object :	a red car
Goal :	to get to his job
Place :	in Paris

Transformers

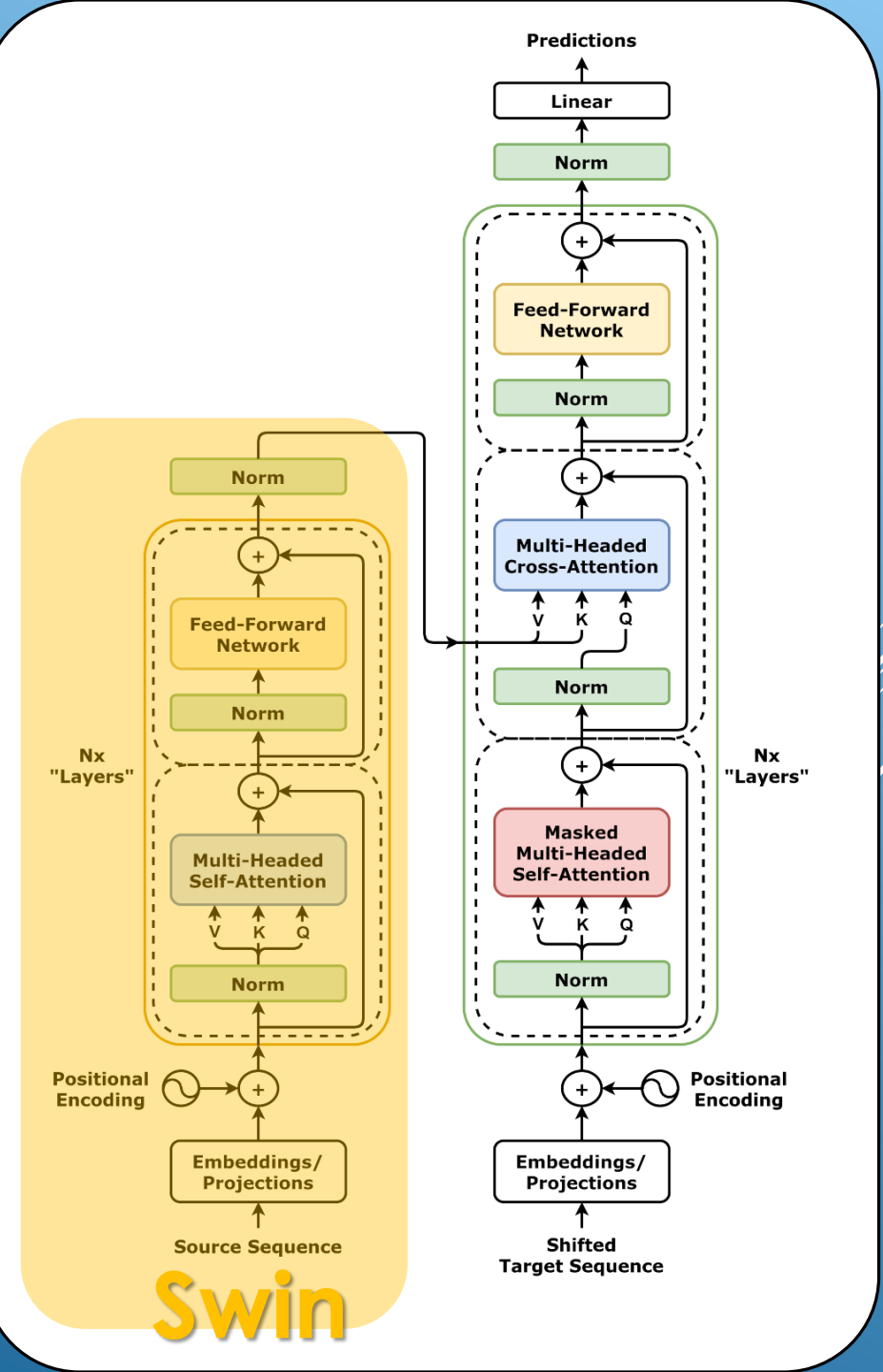


Related tutorials: [Transformers](#)

Transformers

Encoder :

An encoder in AI transforms input data (e.g., text, images) into a compact numerical representation that captures essential features.



Related tutorials: [Transformers](#)

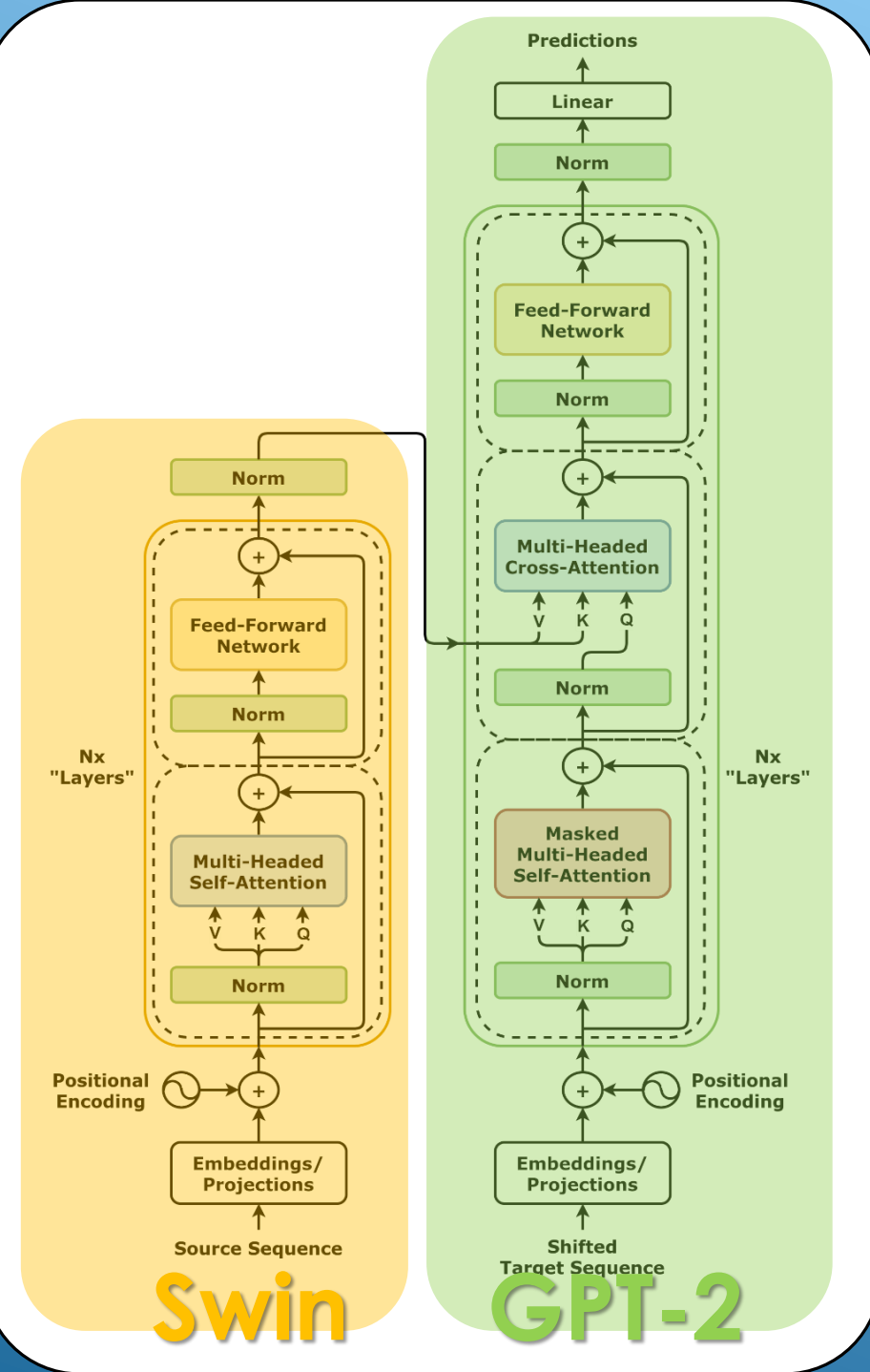
Transformers

Encoder :

An encoder in AI transforms input data (e.g., text, images) into a compact numerical representation that captures essential features.

Decoder :

A decoder in AI takes this encoded representation and reconstructs it back into a human-interpretable format, such as text or images.



Related tutorials: [Transformers](#)

Steps

Step 1: Collect data

Step 2: Chose a relevant architecture

Step 3: Chose metrics to evaluate the model



BLEU metric

Related tutorials: [Introduction to few NLP metrics](#)

BLEU metric

BLEU = BiLingual Evaluation Understudy

We are going to predict mathematical formula

BLEU metric

BLEU = BiLingual Evaluation Understudy

We are going to predict mathematical formula

Is it good?

Goal of a metric:

Evaluate efficiently the performances of the model

BLEU metric

N-gram = sequence of n consecutive words

We are going to predict mathematical formula

BLEU metric

N-gram = sequence of n consecutive words

We are going to predict mathematical formula

1-grams: A diagram illustrating 1-grams. It consists of a horizontal line with several vertical tick marks pointing downwards. There are seven such tick marks, representing individual words in a sequence.

BLEU metric

N-gram = sequence of n consecutive words

We are going to predict mathematical formula

1-grams: 

2-grams: 

BLEU metric

n-gram precision:

$$\frac{\sum_{\mathcal{C} \in \text{unique } n\text{-grams}} \text{count_clip}(\mathcal{C})}{\text{number of } n\text{-grams}}$$

BLEU metric

n-gram precision:

$$\frac{\sum_{\mathcal{C} \in \text{unique } n\text{-grams}} \text{count_clip}(\mathcal{C})}{\text{number of } n\text{-grams}}$$

Low n : Checks the vocabulary of tokens/words used in the candidate

High n : Checks and ensures proper token order and longer syntactic rules

BLEU metric

$$BLEU = BP. \exp \left(\sum_{n=1}^N w_n \log p_n \right)$$

n-gram precision decrease exponentially



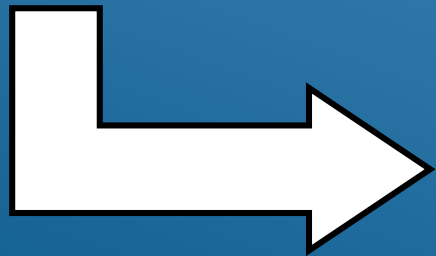
Steps

Step 1: Collect data

Step 2: Chose a relevant architecture

Step 3: Chose metrics to evaluate the model

Step 4: Chose a relevant programming environment



High view library by **Huggingface**: *Transformers, Trainer*
HPC platform: grid5000

Experiments and Results

Training:


- 30 hours on 4 GPUs on grid5000
- We finetuned 600K parameters among 250M parameters
- 250K examples on the training dataset, during 3 epochs

Results:

Model	Google BLEU	Data Size	Parameters
Im2Latex	<i>0.67</i>	441K	243M
TexTeller	<i>0.77</i>	7.5M	300M
Pix2Text	<i>0.07</i>	100K	25M
Sumen	<i>0.47</i>	6.9M	350M
Our	0.23	250K	250M

Experiments and Results

Before training

$$\Delta_+(x, y) = \sum_{n=-\infty}^{\infty} \tilde{\Delta}(x_0 + nT, \vec{x}; y_0, \vec{y})$$


I don't know if it's a coincidence or not," he said. "I think it's just a matter of time before we get to know each other."

He added: "It's not like we're going to be friends. It's just that we're not going to get along very well."

"We're not friends. We're not even friends. I don't think we'll ever get along. I think we're just going to have to get used to each other. We'll just have to figure out how to get through this together. This article is from the archive of our partner .

Experiments and Results

After training

$$\Delta_+(x, y) = \sum_{n=-\infty}^{\infty} \tilde{\Delta}(x_0 + nT, \vec{x}; y_0, \vec{y})$$



`{ \Delta } _ { + } (x , y) = \sum _ { n = - \infty } ^ { \infty } { \tilde { \Delta } (x _ { 0 } + n T , { \vec { x } } ; y _ { 0 } , { \vec { y } }) } . \backslash , | { \% 1 2 3 5 6 7 8 9 0 4 ^ { () } x p = y d e g f a b c n t v q z ~ r s o i u w " j l k > / V _ I + J - : } \backslash q A`

Experiments and Results

After training

$$\Delta_+(x, y) = \sum_{n=-\infty}^{\infty} \tilde{\Delta}(x_0 + nT, \vec{x}; y_0, \vec{y})$$



```
{ \Delta } _ { + } ( x , y ) = \sum _ { n = - \infty } ^ { \infty } { \tilde { \Delta } ( x _ { 0 } + n T , { \vec { x } } ; y _ { 0 } , { \vec { y } } ) } . \ , \ | \ { \% \ 1 \ 2 \ 3 \ 5 \ 6 \ 7 \ 8 \ 9 \ 0 \ 4 \ ^ \ ( \ ) \ x \ p \ = \ y \ d \ e \ g \ f \ a \ b \ c \ n \ t \ v \ q \ z \ \sim \ r \ s \ o \ i \ u \ w \ " \ j \ l \ k \ > \ / \ \_ \ I \ + \ J \ - \ : \ } \ \backslash \ q \ A
```

Conclusion

- Model generalizing well
Limitations: Training times, computation power
- Notebooks covering all project

Conclusion

- Model generalizing well
Limitations: Training times, computation power
- Notebooks covering all project

Datasets:

Basics

Images

Loading

Building

Conclusion

- Model generalizing well
Limitations: Training times, computation power
- Notebooks covering all project

Datasets:

Basics

Images

Loading

Building

ML:

Uses

Training time

Computer's power

Conclusion

- Model generalizing well
Limitations: Training times, computation power
- Notebooks covering all project

Datasets:

Basics
Images
Loading
Building

ML:

Uses
Training time
Computer's power

Theory:

GAs
Metrics
Datas
Torch
Transformers
Sklearn

Conclusion

- Model generalizing well
Limitations: Training times, computation power
- Notebooks covering all project

Datasets:

Basics
Images
Loading
Building

ML:

Uses
Training time
Computer's power

Theory:

GAs
Metrics
Datas
Torch
Transformers
Sklearn

Finetunning:

Set up
Preprocess
Training
Results

Futur/ next steps

- Improve model performance
- Extend to the PDF to Latex tool
- Provide the associated notebooks

IPDF2LATEX

Website: <https://tutorial-ia-pe.readthedocs.io/en/latest/>

GitHub: <https://github.com/GeckSpy/IPdf2Latex>

Macéo Ottavy, Mathieu Longatte, Louison Mocq, Ankit Gayen

Supervised by **Simon Delamare**

References

Image-to-LaTeX Converter for Mathematical Formulas and Text

Daniil Gurgurov¹ Aleksey Morshnev²

¹Department of Language Science and Technology, Saarland University

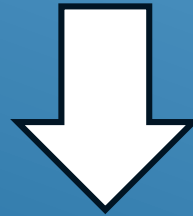
²Department of Data Science and Artificial Intelligence, Saarland University

{dagu00001, almo00008}@stud.uni-saarland.de

Experiments and Results

After training

$$N = \frac{1}{4\beta \left(\frac{2}{\sqrt{d_{\perp}}}\right)^{d_{\perp}}} \int_{y_N}^{y_{end}} \frac{(1-y)^{d_{\perp}-1} dy}{y^3 \left(1 + \left(\frac{d_{\perp}}{4} - \frac{3}{2}\right)y\right)}$$



```
N = \frac { 1 } { \beta \Lambda ^ { d - 1 } } \int _ { \mu _ { N } } ^ { \nu _ { 0 } }
\alpha _ { s } \left( \frac { 2 } { \sqrt { d _ { D } } } \right) ^ { 2 } \over { y ^ { 3 } \left(
1 + \left( 1 + \frac { 3 } { 4 } - \frac { 1 } { 4 } \right) y \right) } } \backslash . \% : = , ( ) 1 2 - 3 4 5 7 8 9 6 |
> { \sim \angle 0 ^ e p a b c d x r v l s i y h t o k q z n I g f }
```