Logbooks & Large Language Models

for accelerator(s)

Antonin Sulc, Raimund Kammering, Annika Eichler, Tim Wilksen
Cape Town,

HELMHOLTZ

DESY.

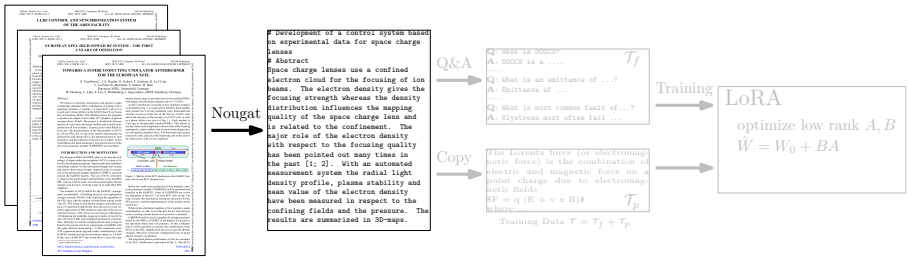
Materials

https://github.com/sulcantonin/WORKSHOP_ICALEPCS23

Introduction

You are about to hear about:

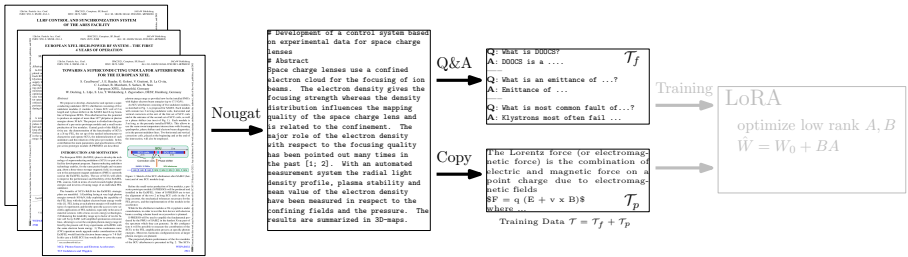
> How to get from a paper to a computer-readable text



Introduction

You are about to hear about:

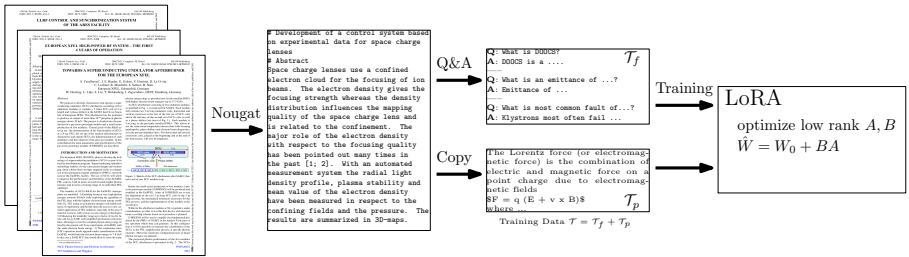
- How to get from a paper to a computer-readable text
- How to make a dataset out of this computer readable text



Introduction

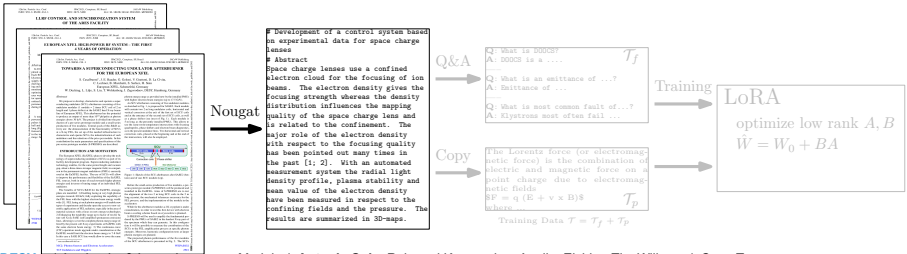
You are about to hear about:

- How to get from a paper to a computer-readable text
- How to make a dataset out of this computer readable text
- How to train a LLM



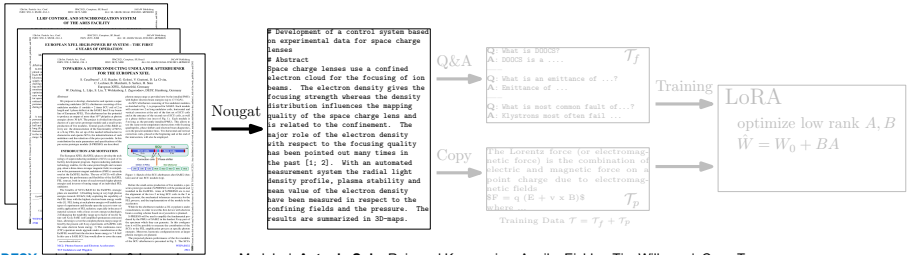
From Anything to Markdown

> There are quite some impressive tools lying around



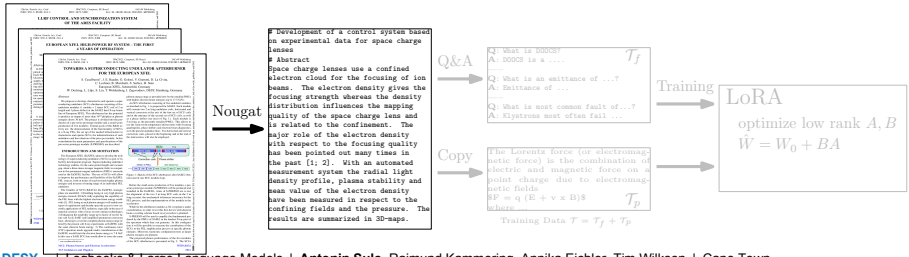
From Anything to Markdown

- There are quite some impressive tools lying around
  - `pdftotext`,



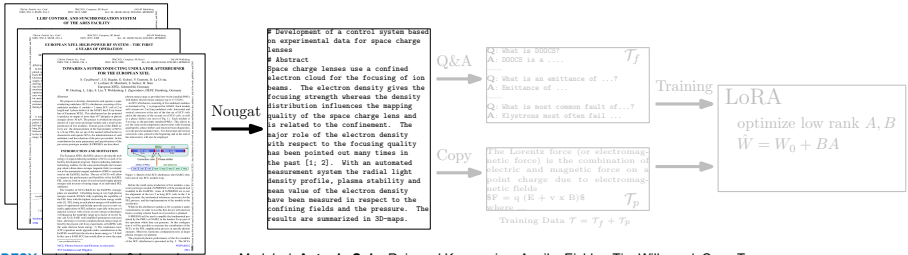
From Anything to Markdown

- There are quite some impressive tools lying around
  - `pdftotext`,
  - `pymupdf`,



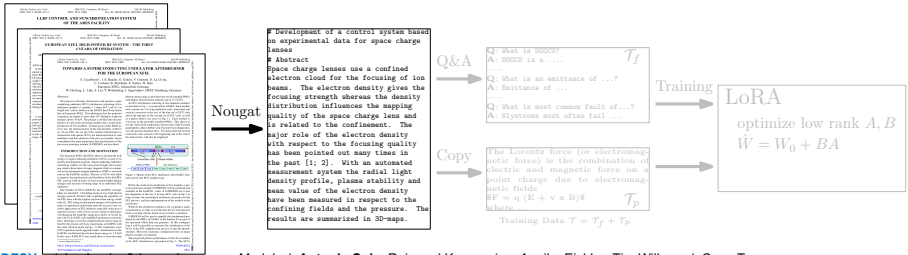
From Anything to Markdown

- There are quite some impressive tools lying around
  - `pdftotext`,
  - `pymupdf`,
  - `detectron`



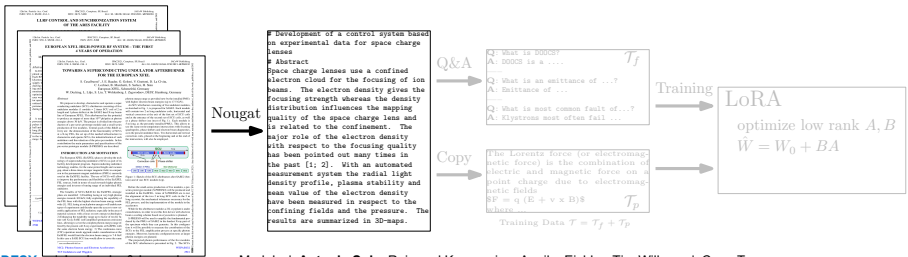
From Anything to Markdown

- There are quite some impressive tools lying around
  - `pdftotext`,
  - `pymupdf`,
  - `detectron`
- These tools are really impressive, only a tiny thing was missing



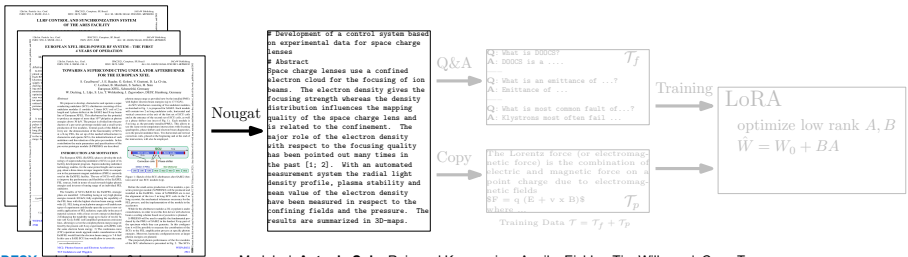
From Anything to Markdown

- There are quite some impressive tools lying around
  - `pdftotext`,
  - `pymupdf`,
  - `detectron`
- These tools are really impressive, only a tiny thing was missing
- **Tables** and **formulas**, so native for scientists



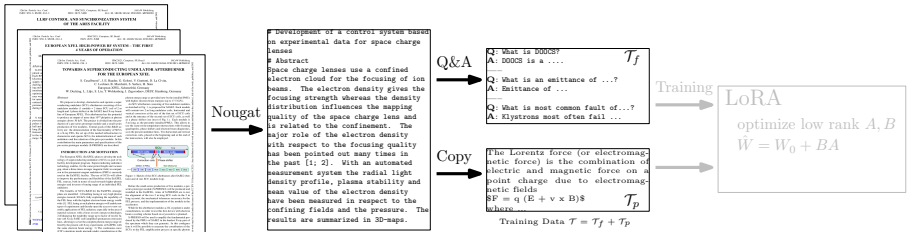
From Anything to Markdown

- There are quite some impressive tools lying around
  - `pdftotext`,
  - `pymupdf`,
  - `detectron`
- These tools are really impressive, only a tiny thing was missing
- **Tables** and **formulas**, so native for scientists
- **Nougat library** (PDF to Markdown), **pandoc** (e.g. LaTeX to Markdown)



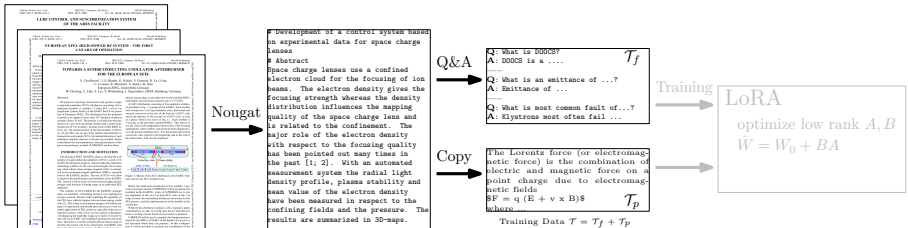
How to Make a Dataset

> Training a LLM requires data



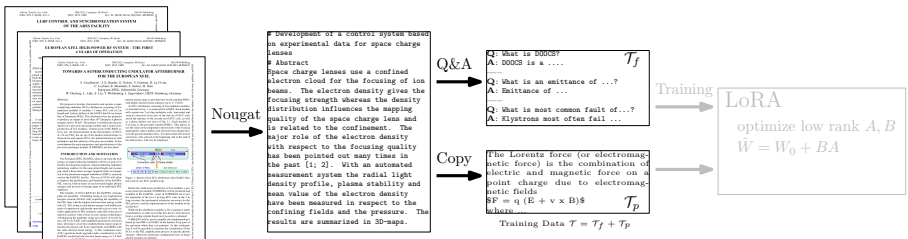
How to Make a Dataset

- Training a LLM requires data
- These data can be unsupervised, meaning training the LLM to predict token (GPT) or fill gap (BERT)



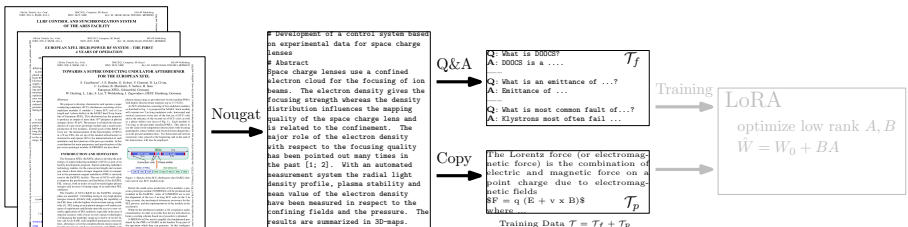
How to Make a Dataset

- Training a LLM requires data
- These data can be unsupervised, meaning training the LLM to predict token (GPT) or fill gap (BERT)
- However, the most important are supervised data, LLM is asked questions, you need to provide Q&A pairs



How to Make a Dataset

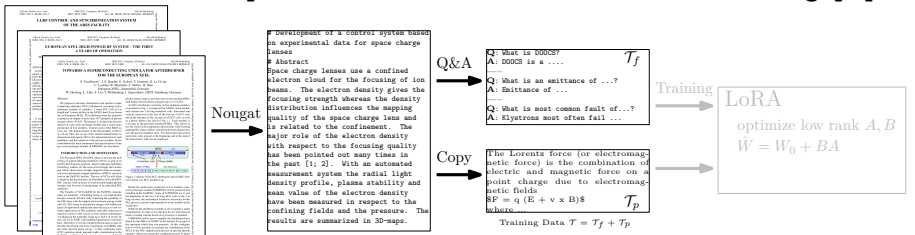
- Training a LLM requires data
- These data can be unsupervised, meaning training the LLM to predict token (GPT) or fill gap (BERT)
- However, the most important are supervised data, LLM is asked questions, you need to provide Q&A pairs
- Creating a Q&A (supervised) dataset can be simulated



How to Make a Dataset

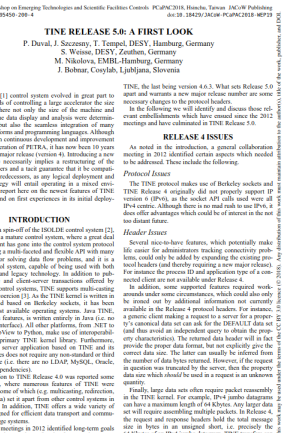
- Training a LLM requires data
- These data can be unsupervised, meaning training the LLM to predict token (GPT) or fill gap (BERT)
- However, the most important are supervised data, LLM is asked questions, you need to provide Q&A pairs
- Creating a Q&A (supervised) dataset can be simulated
- A pre-trained LLM can do many things, one of them is generating questions.

```
"Generate 10 questions with answers for a following paper:$PAPER"
```



Example

TINE RELEASE 5.0: A FIRST LOOK



12th Int. Workshop on Emerging Technologies and Scientific Facilities Controls PCaPAC2018, Hsinchu, Taiwan JACoW Publishing
ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP19

TINE RELEASE 5.0: A FIRST LOOK

P. Duval, J. Szczesny, T. Tempel, DESY, Hamburg, Germany
S. Weisse, DESY, Zeuthen, Germany
M. Nikolova, EMBL-Hamburg, Germany
J. Bobnar, Cosylab, Ljubljana, Slovenia

Abstract

The TINE [1] control system evolved in great part to meet the needs of controlling a large accelerator the size of HERA, where not only the size of the machine and efficient online data display and analysis were determining criteria, but also the seamless integration of many different platforms and programming languages. Although there has been continuous development and improvement during the operation of PETRA, it has now been 10 years since the last major release (version 4). Introducing a new major release necessarily implies a restructuring of the protocol headers and a tacit guarantee that it be compatible with its predecessors, as any logical deployment and upgrade strategy will entail operating in a mixed environment. We report here on the newest features of TINE Release 5.0 and on first experiences in its initial deployment.

INTRODUCTION

Originally a spin-off of the ISOLDE control system [2], TINE is both a mature control system, where a great deal of development has gone into the control system protocol itself, offering a multi-faceted and flexible API with many alternatives for solving data flow problems, and it is a modern control system, capable of being used with both cutting-edge and legacy technology. In addition to publish-subscribe and client-server transactions offered by many other control systems, TINE supports multi-casting and contract coercion [3]. As the TINE kernel is written in straight C and based on Berkeley sockets, it has been ported to most available operating systems. Java TINE, with all of its features, is written entirely in Java (i.e. no Java Native Interface). All other platforms, from .NET to Matlab to LabView to Python, make use of interoperability with the primary TINE kernel library. Furthermore, any client or server application based on TINE and its central services does not require any non-standard or third party software (i.e. there are no LDAP, MySQL, Oracle, Log4j, etc. dependencies).

The transition to TINE Release 4.0 was reported some time ago [4], where numerous features of TINE were enumerated, some of which (e.g. multicasting, redirection, structured data) set it apart from other control systems in common use. In addition, TINE offers a wide variety of features designed for efficient data transport and communication in large systems.

A series of meetings in 2012 identified long-term goals and established a roadmap for the future Release 5.0. Many of these goals have been realized over the past several years, showing up in new minor release versions of TINE, the last being version 4.6.3. What sets Release 5.0 apart and warrants a new major release number are some necessary changes to the protocol headers.

In the following we will identify and discuss these relevant embellishments which have ensued since the 2012 meetings and have culminated in TINE Release 5.0.

RELEASE 4 ISSUES

As noted in the introduction, a general collaboration meeting in 2012 identified certain aspects which needed to be addressed. These include the following:

Protocol Issues

The TINE protocol makes use of Berkeley sockets and TINE Release 4 originally did not properly support IP version 6 (IPv6), as the socket API calls used were all IPv4 centric. Although there is no mad rush to use IPv6, it does offer advantages which could be of interest in the not too distant future.

Header Issues

Several nice-to-have features, which potentially make life easier for administrators tracking connectivity problems, could only be added by expanding the protocol headers (and thereby requiring a new major release). For instance the process ID and application type of a connected client are not available under Release 4.

In addition, some supported features required workarounds under some circumstances, which could also only be ironed out by additional information not currently available in the Release 4 protocol headers. For instance, a generic client making a request to a server for a property's canonical data set can ask for the DEFAULT data set (and thus avoid an independent query to obtain the property characteristics). The returned data header will in fact provide the proper data format, but not explicitly give the correct data size. The latter can usually be inferred from the number of data bytes returned. However, if the request in question was truncated by the server, then the property's data size which *should* be used in a request is an unknown quantity.

Finally, large data sets often require packet reassembly in the TINE kernel. For example, IP+4 jumbo datagrams can have a maximum length of 64 Kbytes. Any larger data set will require assembling multiple packets. In Release 4, the request and response headers hold the total message size in bytes in an unsigned short, i.e. precisely the 64 Kbytes of an IPv4 jumbo datagram. TINE transfers can of course use a TCP stream, or shared memory, rather than datagrams, but the same packet reassembly exists.

- Q: What is TINE control system?
- A: TINE is a control system that evolved from the ISOLDE control system...
- Q: What are the new features in TINE Release 5.0?'
- A: The new features ...
- Q: What are the advantages of using IPv6 in TINE?
- A: IPv6 offers advantages such as larger data sets that can be transferred without packet reassembly, jumbo datagrams up...

How to train LLM

> Fine-tuning LLM is **very costy**, if you want to opitmize a parameter, a gradient of that parameter is calculated (w.r.t. a loss function)

How to train LLM

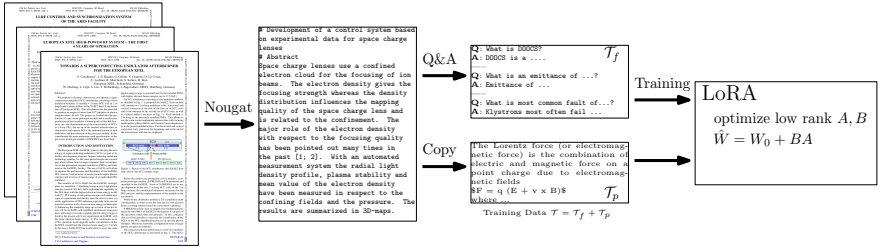
- Fine-tuning LLM is **very costy**, if you want to opitmize a parameter, a gradient of that parameter is calculated (w.r.t. a loss function)
- This can be quite explosive, because of **chain-rule**.

How to train LLM

- Fine-tuning LLM is **very costy**, if you want to opitmize a parameter, a gradient of that parameter is calculated (w.r.t. a loss function)
- This can be quite explosive, because of **chain-rule**.
- There are parameter efficient workarounds, like **LoRA** (Low-Rank Adaptation)

How to train LLM

- Fine-tuning LLM is **very costy**, if you want to opitmize a parameter, a gradient of that parameter is calculated (w.r.t. a loss function)
- This can be quite explosive, because of **chain-rule**.
- There are parameter efficient workarounds, like **LoRA** (Low-Rank Adaptation)
- Consider that you have a parameter matrix W, instead of trying to find ∇W, you are optimizing two low rank matrices B and A, which you add to the original (fixed) W_0, i.e.



Live Demo

Thank you!

Contact

Deutsches Elektronen-
Synchrotron DESY

www.desy.de

Antonin Sulc, Raimund Kammering, Annika Eichler, Tim Wilksen
0000-0001-7767-778X
MCS
antonin.sulc@desy.de