Hands-On Session

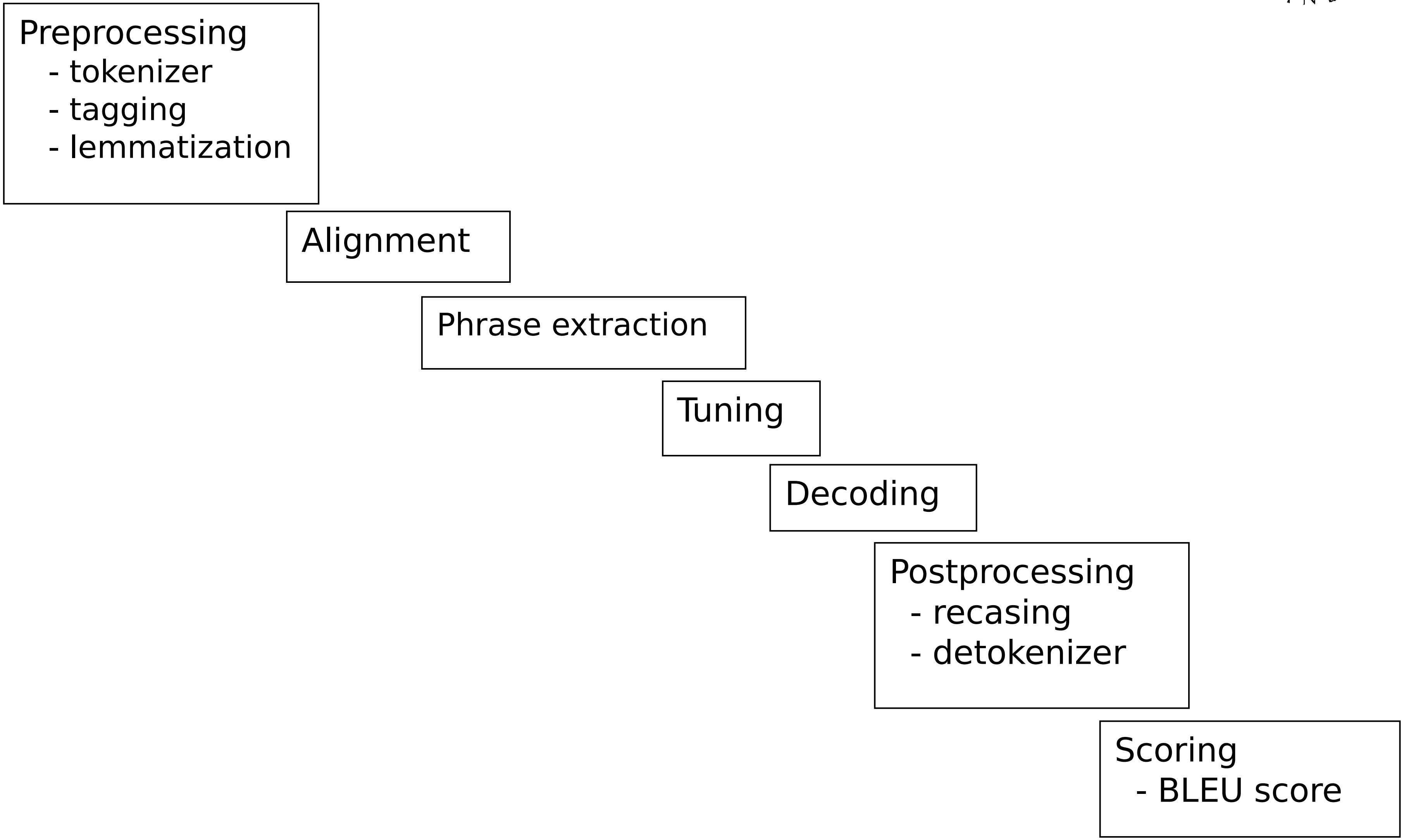
Hieu Hoang

Matthias Huck

October 2014

THE UNIVERSITY OF EDINBURGH

SMT Pipeline



Preprocessing
- tokenizer
- tagging
- lemmatization

Alignment

Phrase extraction

Tuning

Decoding

Postprocessing
- recasing
- detokenizer

Scoring
- BLEU score

Using the Experiment Management System (EMS)

- In brief
  - from raw data to tuned MT system
- Wrapper for everything needed to
  - Train
  - Tuning
  - Decode
  - Evaluate
- Require
  - config file

Using the EMS

ssh guest@odin.inf.ed.ac.uk

Password = Edinburgh123

cd workspace/experiment/fr-en/<river>/

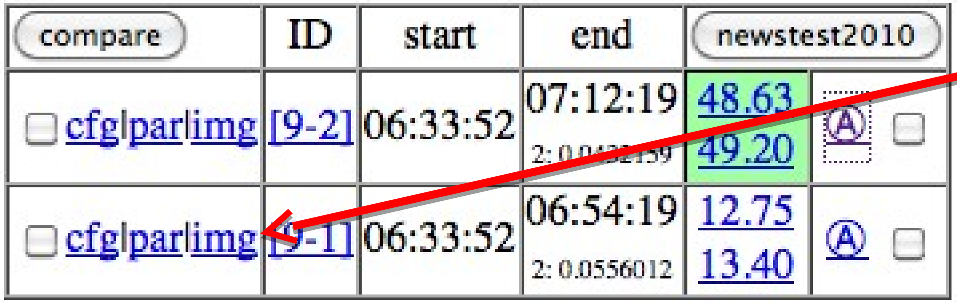
nohup ./run.new.sh config.pb &

config file:

steps/1/config.1

Browse: http://amta.statmt.org/

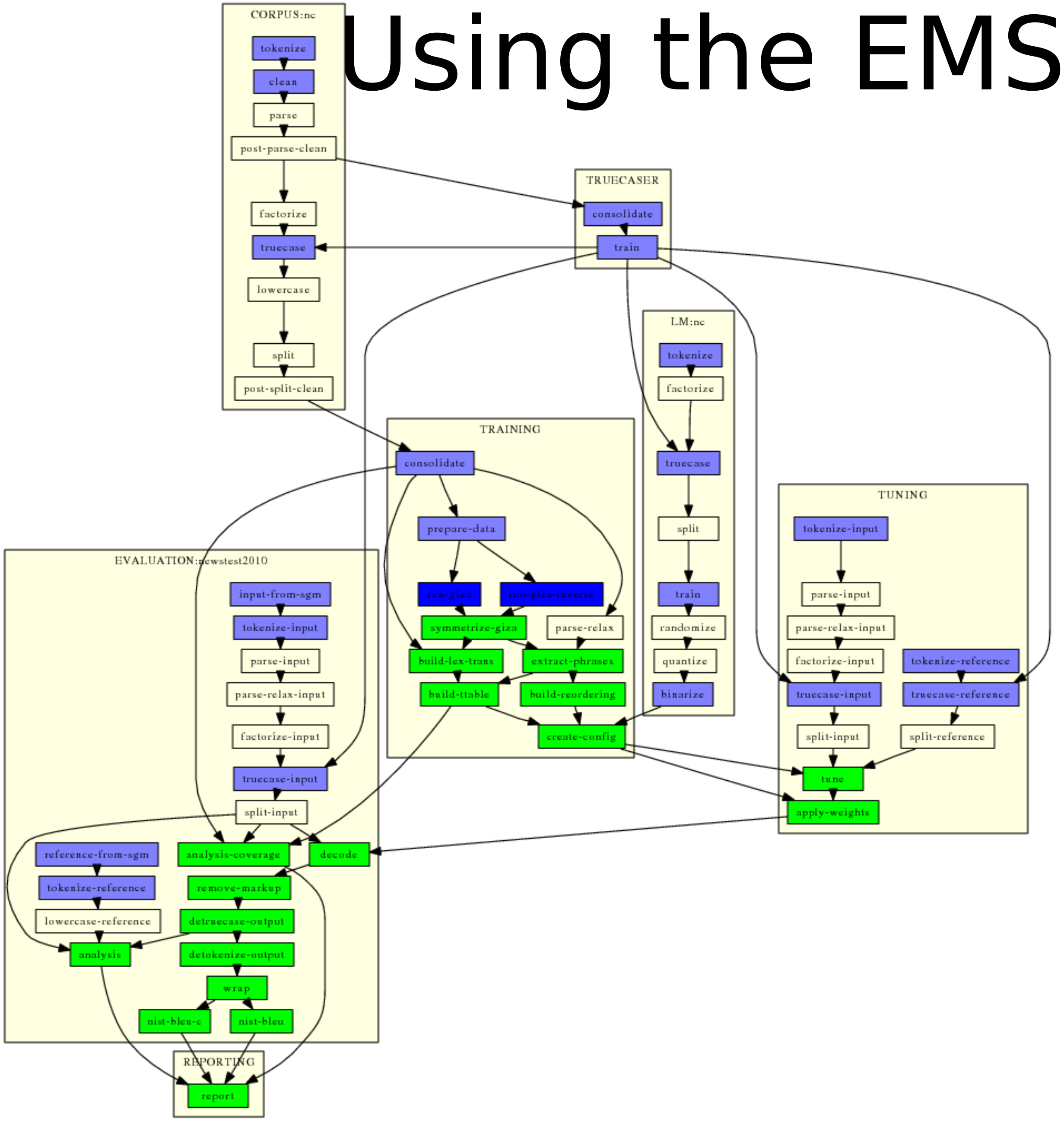
- find your experiment



compare	ID	start	end	newstest2010		
□ cfg|par|img	[9-2]	06:33:52	07:12:19 2: 0.0432159	48.63 49.20	Ⓐ	□
□ cfg|par|img	[9-1]	06:33:52	06:54:19 2: 0.0556012	12.75 13.40	Ⓐ	□

Click Here

Using the EMS



Hierarchical/Syntax Models

- Same pipeline
- Different
  - Extraction
  - Decoding
- Reuse common output
  - Data cleaning
  - Tokenization
  - Alignment

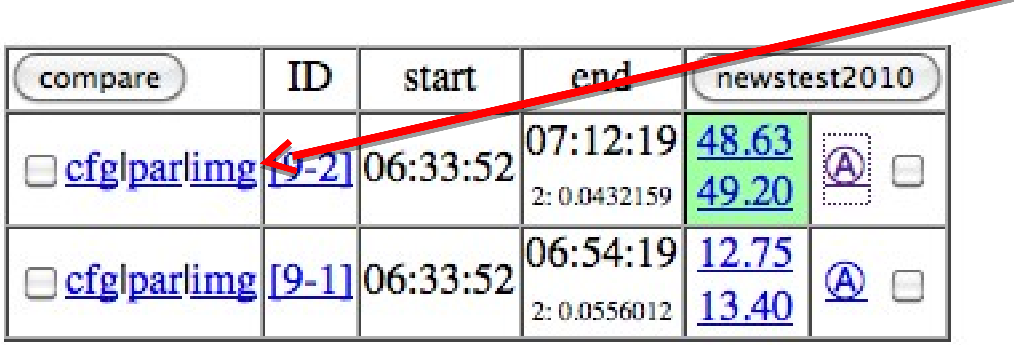
Hierarchical/Syntax Models

```
nohup ./run.new config.hiero &
```

Browse: http://amta.statmt.org/

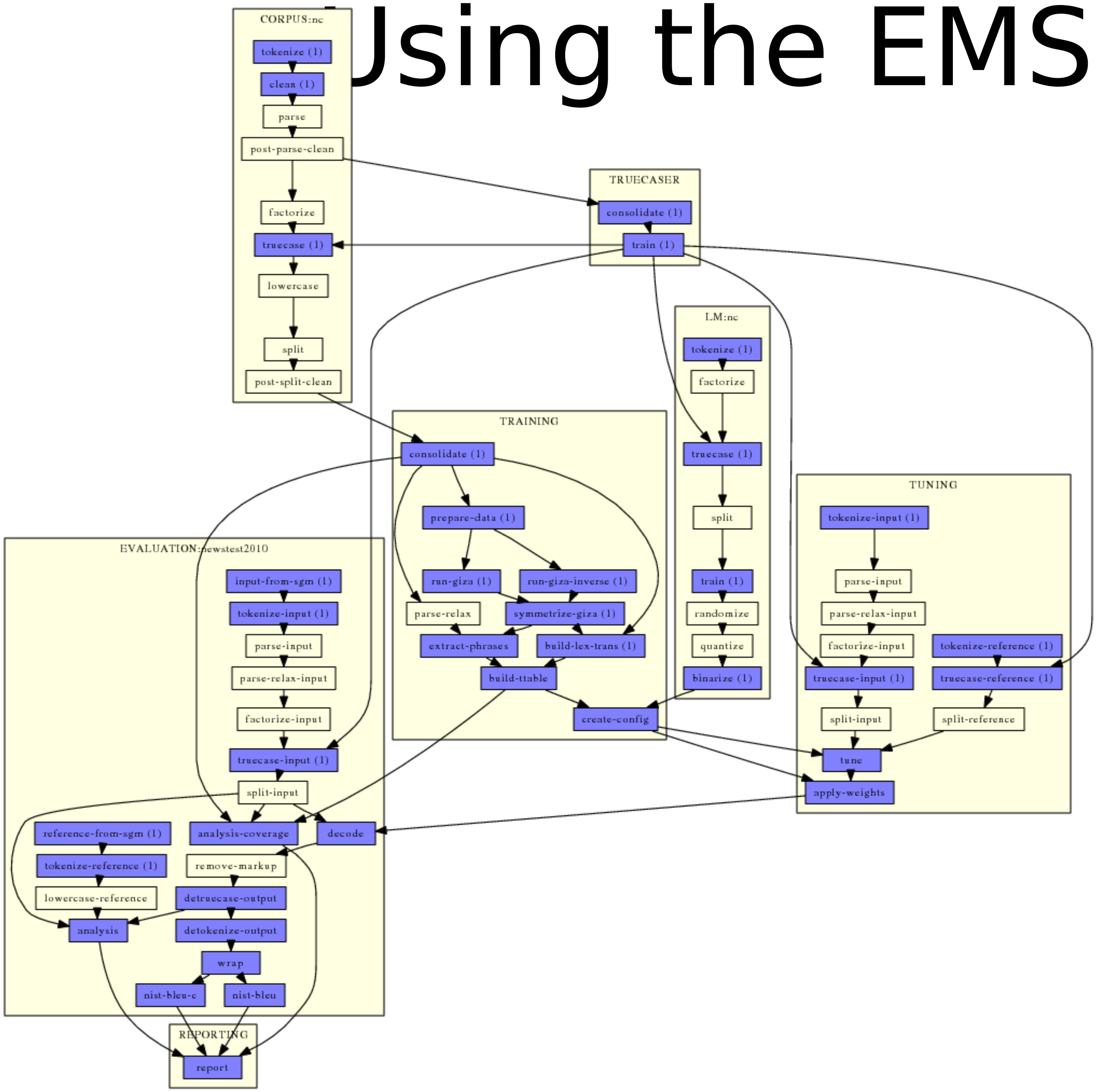
- find your experiment

Click Here

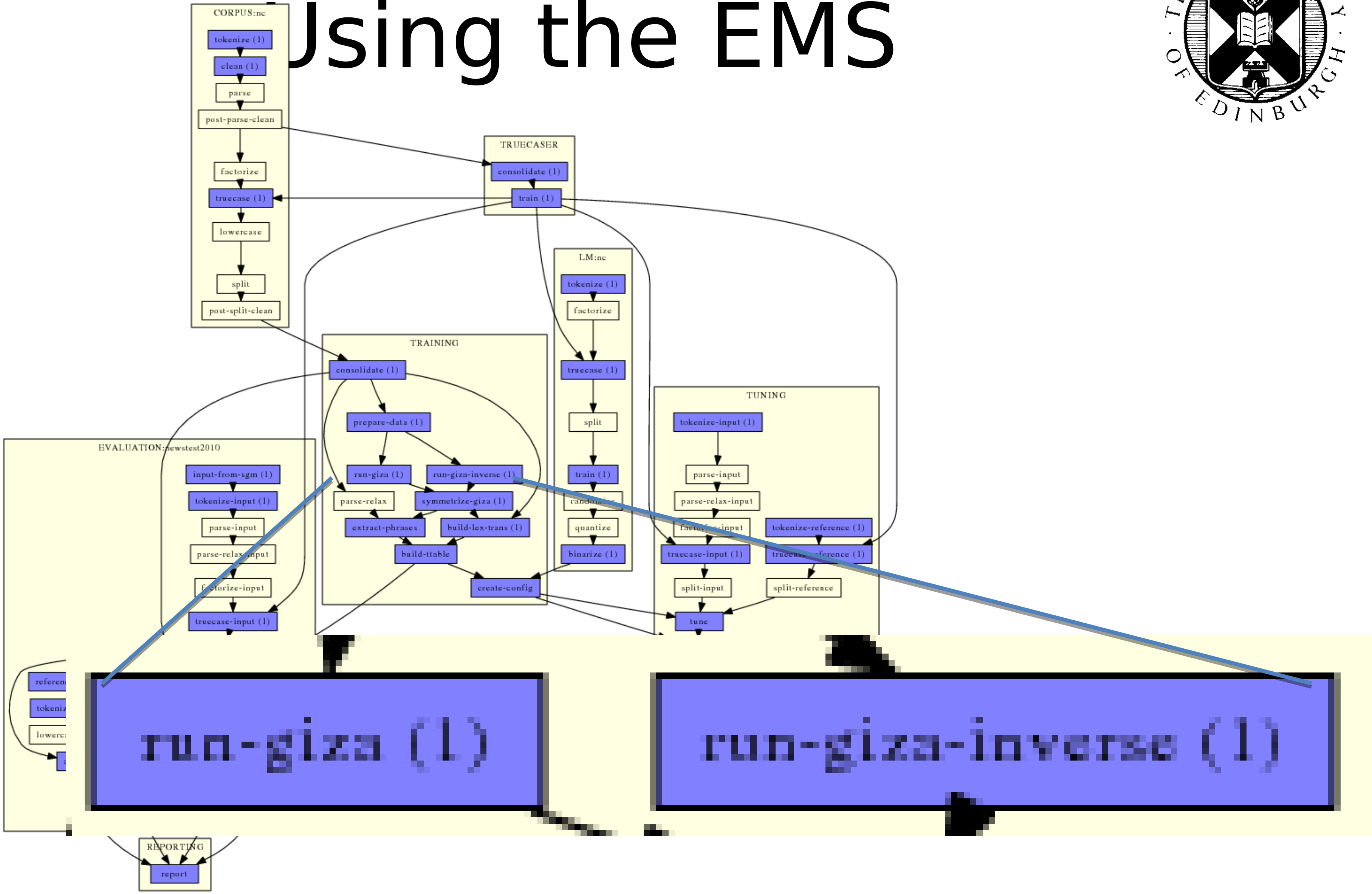


compare	ID	start	end	newstest2010	
□ cfg|par|img	[9-2]	06:33:52	07:12:19 2: 0.0432159	48.63 49.20	Ⓐ □
□ cfg|par|img	[9-1]	06:33:52	06:54:19 2: 0.0556012	12.75 13.40	Ⓐ □

Using the EMS



Using the EMS



Using the EMS

- Standardized directories
  - corpus
  - evaluation
  - lm
  - model
  - steps
  - training
  - recasing
  - Tuning
- Standardized file names for each experiments, eg.
  - nc.lm.1 - nc.lm.2
  - extract.1 - extract.1
  - moses.ini.1 - moses.ini.2

Debugging

- Incorrect file paths
  - Data files
  - Executables
- Bugs in scripts
- Bugs in data

Debugging

- steps /<num>/
  - Working directory of EMS
- config.<num>
  - Parameters for experiment <num>
- Scripts that run moses script
  - eg. TRAINING_run_giza.1
- Error logs produced by every script
  - eg. TRAINING_run_giza.1.STDERR

Summary of EMS

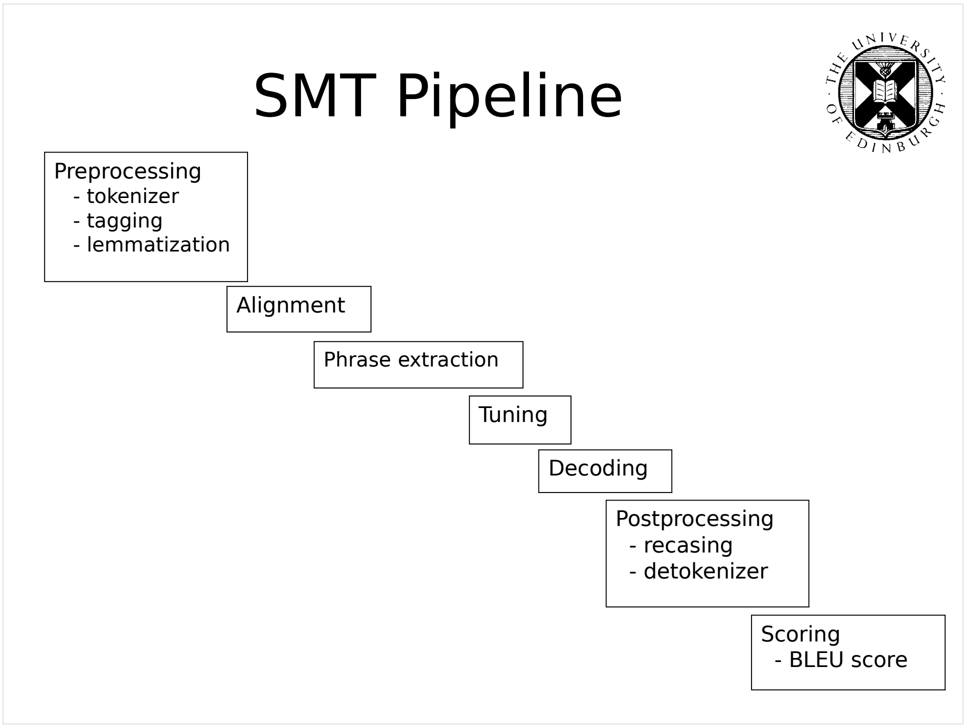
- Create complete MT system from raw data
- Consistent, reproduceable
- Efficient
  - Re-use when it can
- Analysis
- Support parallelization
  - SGI
  - Multi-threading

Hands-On Session

Hieu Hoang

Matthias Huck

October 2014



SMT Pipeline

Preprocessing
- tokenizer
- tagging
- lemmatization

Alignment

Phrase extraction

Tuning

Decoding

Postprocessing
- recasing
- detokenizer

Scoring
- BLEU score

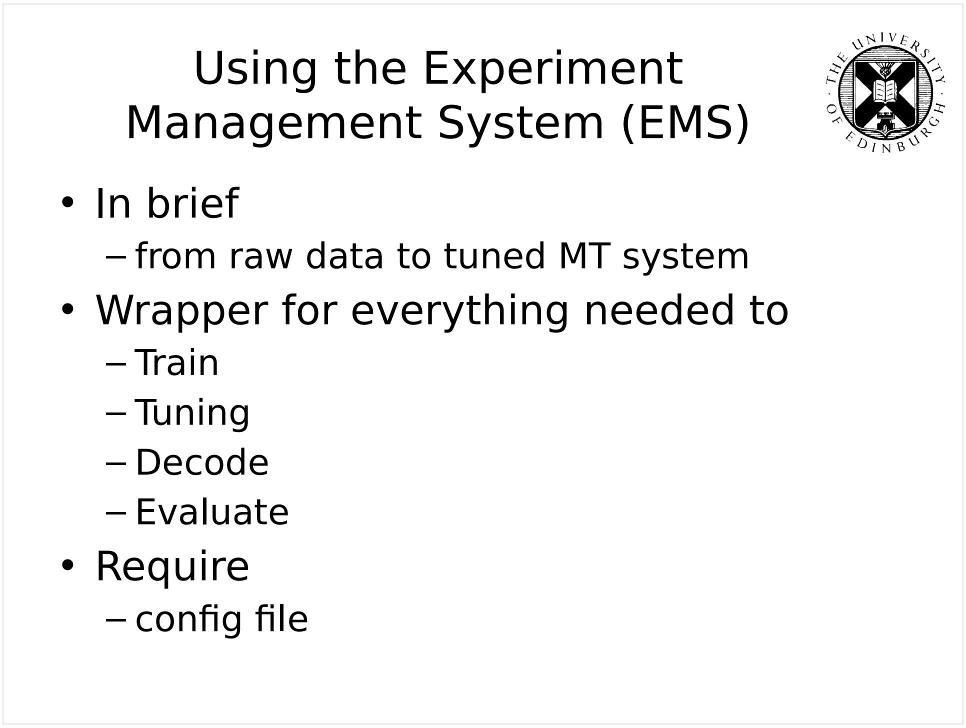
MT pipeline
- each part is critical to producing good MT system

Can show you how to do each part
- take a week

Lose the will to live!

However, not necessary to know the mechanics of each & every part to start

Those that don’t need to know, or know but just want it to work consistently
- provide a system which wraps up the pipeline



Using the Experiment Management System (EMS)

- In brief
  - from raw data to tuned MT system
- Wrapper for everything needed to
  - Train
  - Tuning
  - Decode
  - Evaluate
- Require
  - config file

This system is called the EMS

- included with the Moses toolkit
- a script that creates a series of scripts

What does EMS do?

- takes raw data
- turn into a phrase-based system, or hierarchical system

By running the entire pipeline.

All you need to do it give is a config file

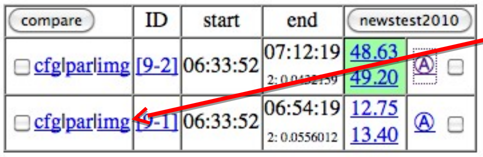
Using the EMS

```
ssh guest@odin.inf.ed.ac.uk
   Password = Edinburgh123

cd workspace/experiment/fr-en/<river>/
 nohup  ./run.new.sh config.pb &
```

config file:
  steps/1/config.1

Browse: http://amta.statmt.org/
  - find your experiment



compare	ID	start	end	newstest2010		
□ cfg|par|img	[9-2]	06:33:52	07:12:19 2: 0.0432159	48.63 49.20	Ⓐ	□
□ cfg|par|img	[9-1]	06:33:52	06:54:19 2: 0.0556012	12.75 13.40	Ⓐ	□

Click Here

We’ve set up a small experiemtn for you
  - so log onto the our server as guest
  - password is ‘welcome’
And run this script
  - it’s a standard phrase-based system. Training on just 1000 lines of data
  - hopefully it’ll finish in 10-15 minutes.

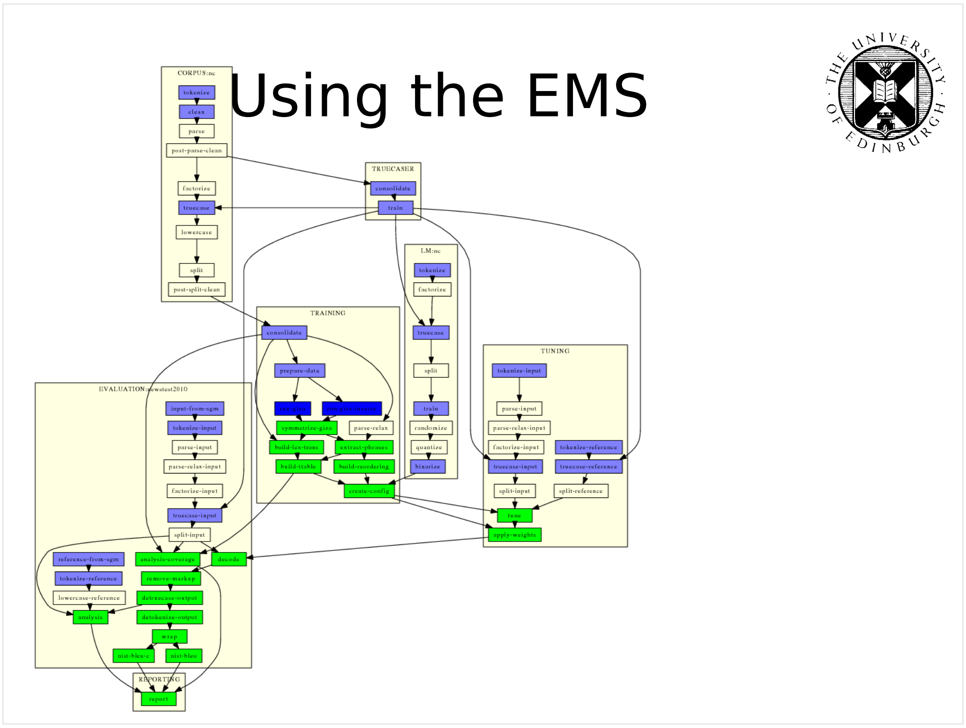
  - can look at this script, it’s 1 line which runs the EMS, telling it which config file to use, in this case it says the 1st config file.

So where’s the config file?
  - it’s in the steps directory, under ‘1’
  - look inside it

  - doesn’t sayhow things are run, EMS already knows that
  - instead, specify where the data and the executables are found
  - <PAUSE> for questions

Now it’s running, can check on it progress online.

Go to this URL with firefox or whatever
  - find your experiment. (Name of fruit)
  - ‘cfg’ is the config file your supplied it with
  - par is the parameters the EMS extracted from the config file
  - click on the ‘image’ link
    - shows you progress of your running experiment



Using the EMS

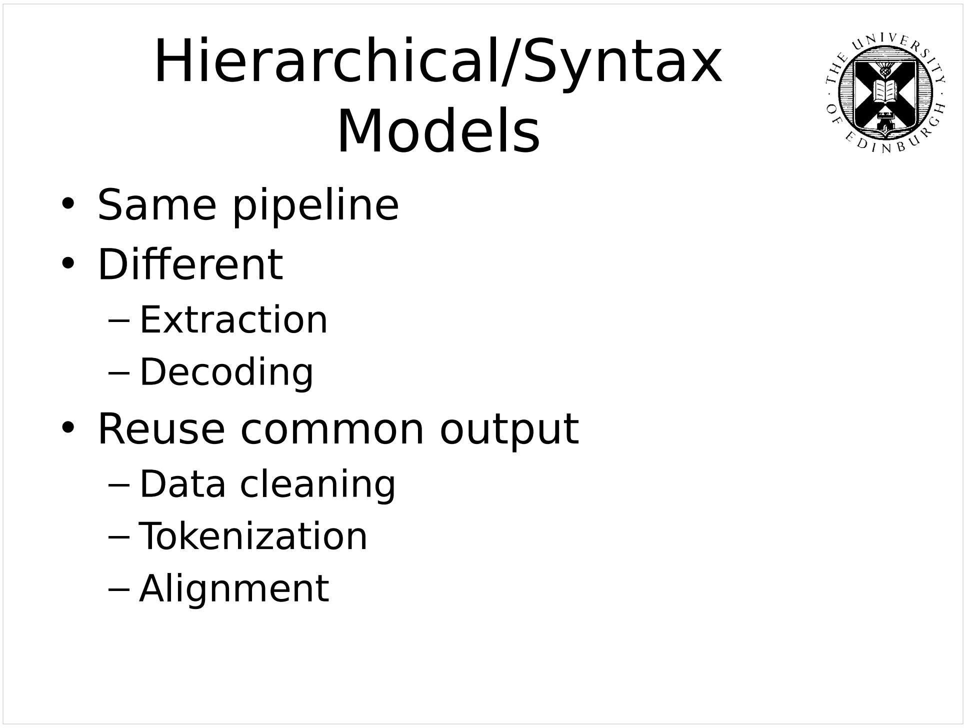
All the steps that you have to do to create MT systems

Refresh your browser to see updates

Wait for all boxes to turn light-blue. Everything has finished!

Green – still to do
Dark blue – Running jobs
- in this 1, GIZA and inverse giza is still running.

If you see RED
- that means that step broke. You have to fix it
R

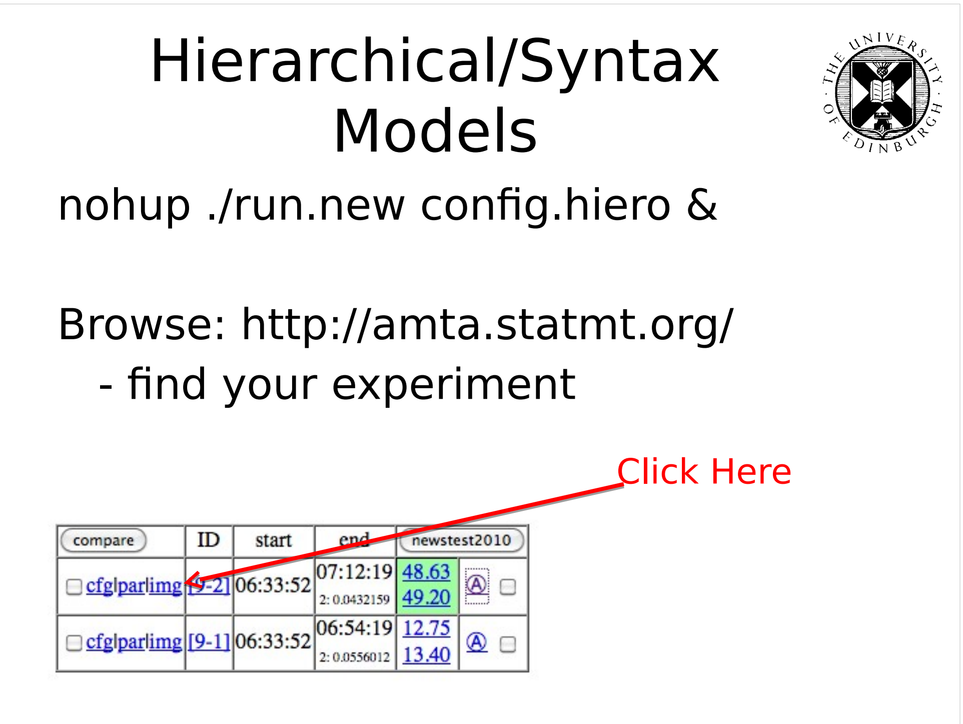


Hierarchical/Syntax Models

- Same pipeline
- Different
  - Extraction
  - Decoding
- Reuse common output
  - Data cleaning
  - Tokenization
  - Alignment

For hierarchical and syntax models, only the extraction and decoding is different.

Many of the other steps are identical. The EMS understand that
- if a step has identical input and output, and the same arguments
- reuse



Hierarchical/Syntax Models

```
nohup ./run.new config.hiero &
```

Browse: http://amta.statmt.org/
- find your experiment

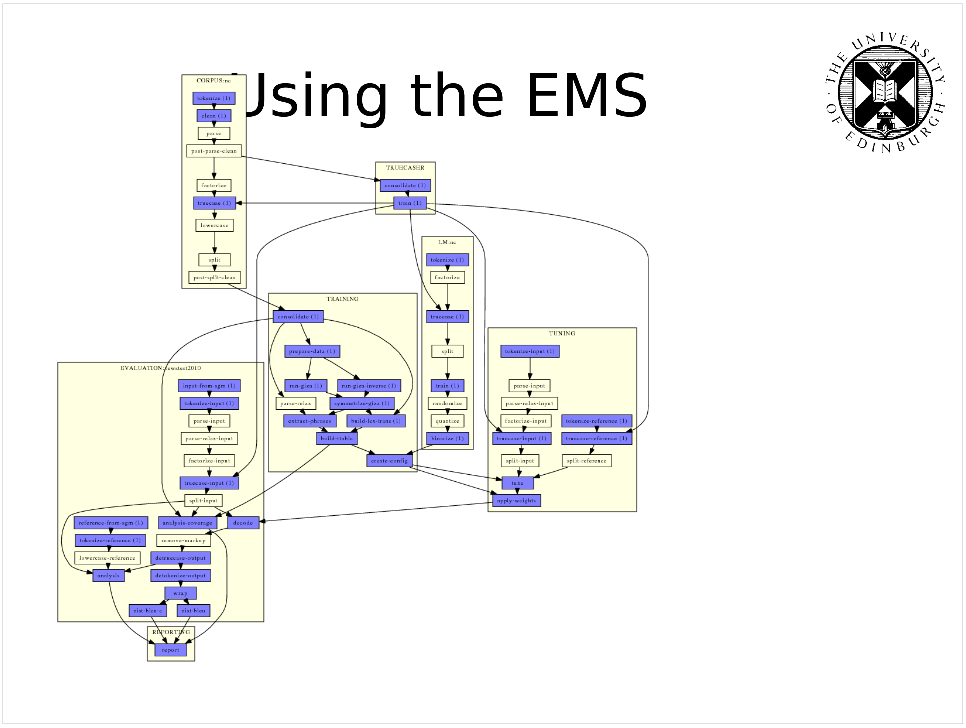
Click Here

compare	ID	start	end	newstest2010		
□ cfg|par|img	[9-2]	06:33:52	07:12:19 2: 0.0432159	48.63 49.20	Ⓐ	□
□ cfg|par|img	[9-1]	06:33:52	06:54:19 2: 0.0556012	12.75 13.40	Ⓐ	□

So do the same thing for another experiement

This uses the same data
- but creates a hierarchical model instead of a phrase-based model

Find your experiment again

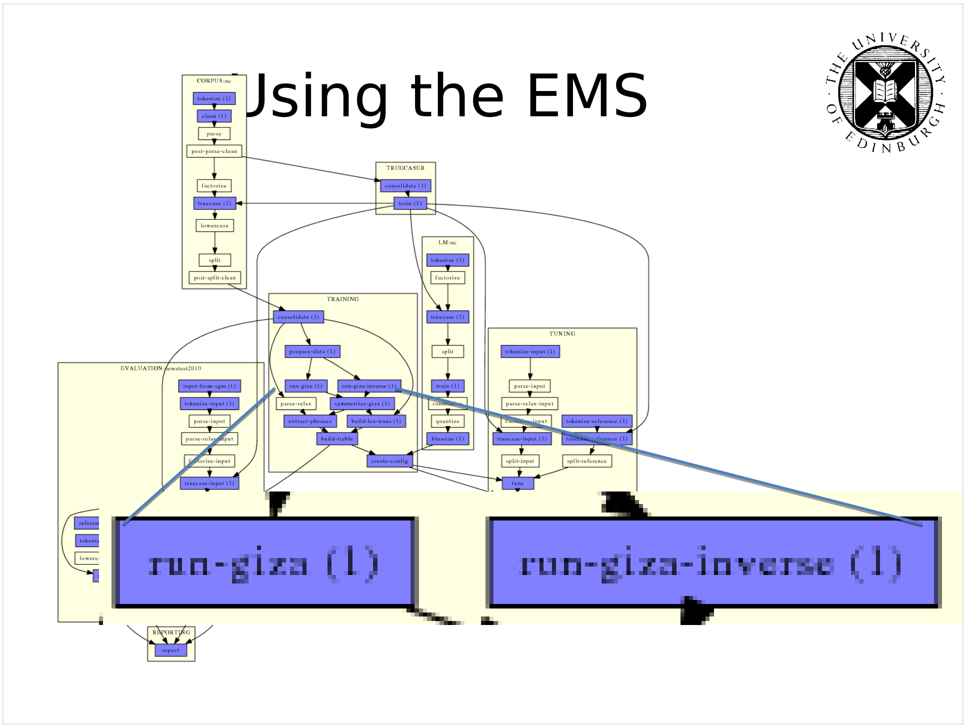


Using the EMS

Looks exactly like the phrase-based

However, those parts that can be resused from the phrase-based
- aren’t green, or blue
- those steps are white
    - meaning the EMS will reuse the output from the phrase-based experiment instead.

- steps like tokenization, cleaninng, alignment are all white



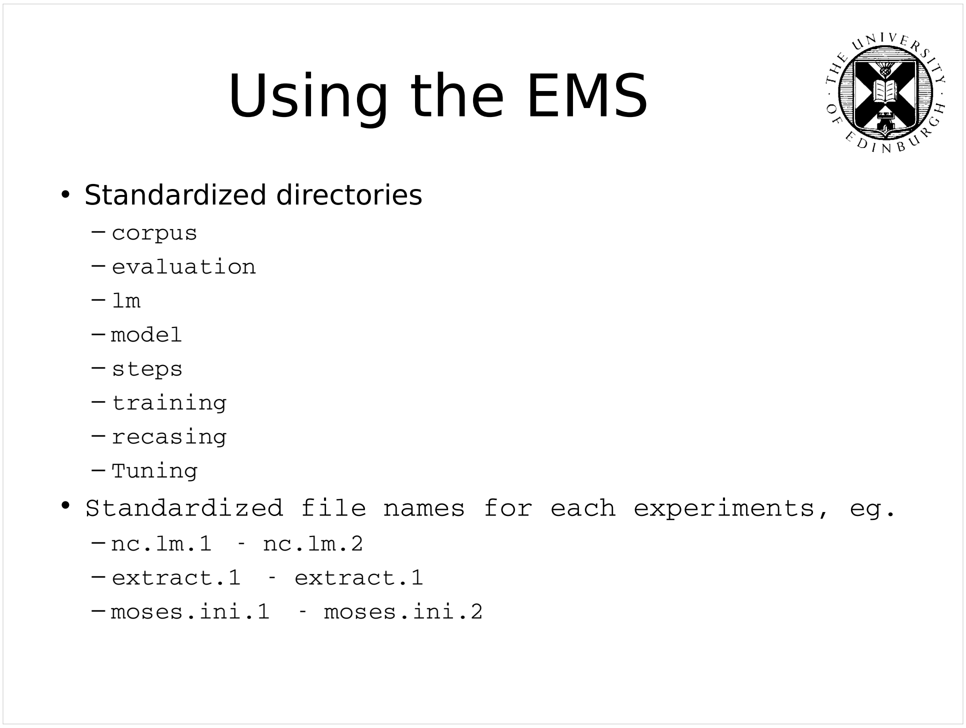
Using the EMS

Looks exactly like the phrase-based

However, those parts that can be resused from the phrase-based

- aren’t green, or blue
- those steps are white
    - meaning the EMS will reuse the output from the phrase-based experiment instead.

- steps like tokenization, cleaninng, alignment are all white



Using the EMS

- Standardized directories
  - corpus
  - evaluation
  - lm
  - model
  - steps
  - training
  - recasing
  - Tuning
- Standardized file names for each experiments, eg.
  - nc.lm.1 - nc.lm.2
  - extract.1 - extract.1
  - moses.ini.1 - moses.ini.2

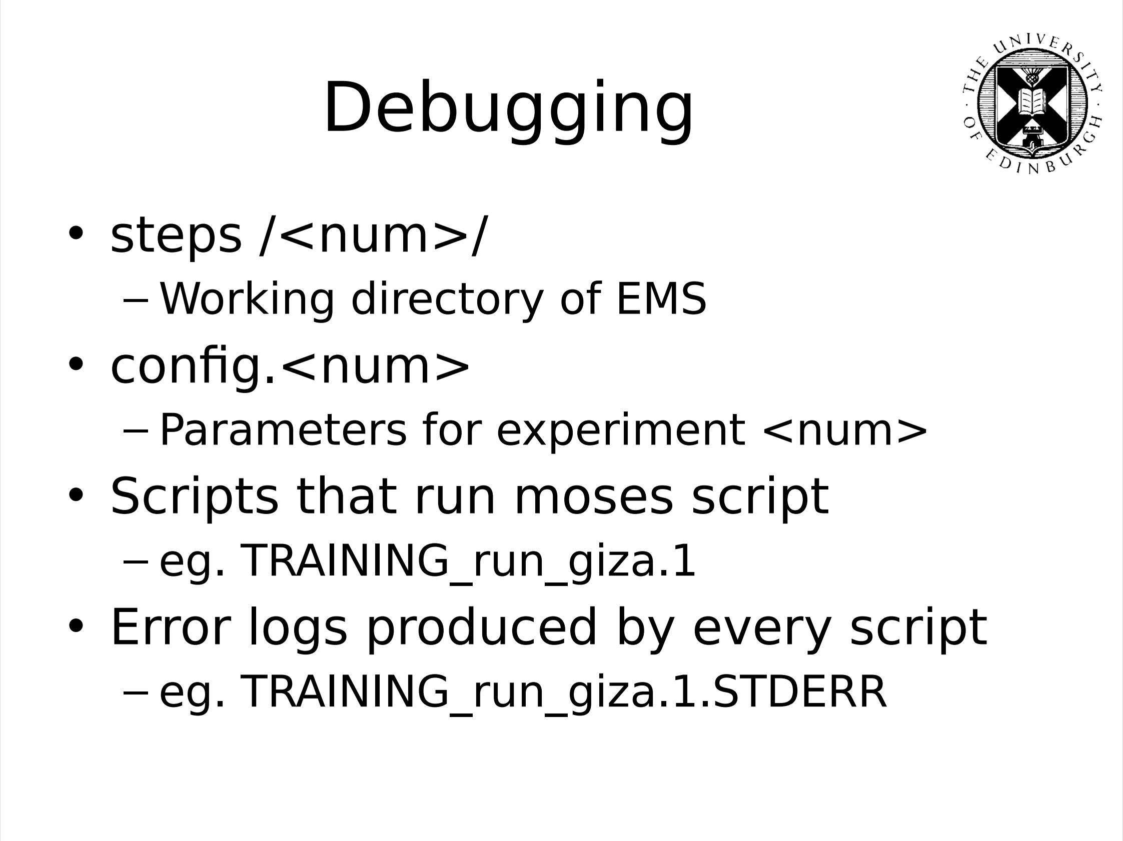
Digging deeper into how the EMS works

Standardized directories

Standardized naming convention for files
- easy to see which exactly which files have been created by which experiment

Debugging

- Incorrect file paths
  - Data files
  - Executables
- Bugs in scripts
- Bugs in data



Debugging

- steps /<num>/
  - Working directory of EMS
- config.<num>
  - Parameters for experiment <num>
- Scripts that run moses script
  - eg. TRAINING_run_giza.1
- Error logs produced by every script
  - eg. TRAINING_run_giza.1.STDERR

Summary of EMS

- Create complete MT system from raw data
- Consistent, reproduceable
- Efficient
  - Re-use when it can
- Analysis
- Support parallelization
  - SGI
  - Multi-threading