

iEMS – Interactive Experiment Management System

Final Report

Pēteris Nikiforovs

Introduction

Interactive Experiment Management System (Interactive EMS or iEMS) is an experiment management system with a graphical user interface for designing and running statistical machine translation experiments.

iEMS is an environment where it is possible to design and train a complete machine translation engine from scratch. There is an optional backend that can be used to launch and configure virtual servers on the Amazon Web Services cloud for running experiments.

It is written in JavaScript and runs in all modern desktop browsers with no installation.

It is an open-source project licensed under the Apache 2.0 license.

There was a rough prototype developed at Machine Translation Marathon 2015 in Prague, Czech Republic. The aims of this project submitted to EAMT are to improve iEMS and create new features.

Milestones and deliverables

The project source code is freely available on GitHub at <https://github.com/pdonald/iems>

There is a demo video available on YouTube <https://www.youtube.com/watch?v=BRu9kYJ9-iE> It showcases the following functionality:

- creating a new MT experiment for the English-Latvian language pair
 - European Constitution parallel corpus from OPUS (automatically downloaded)
 - tokenization using Moses tokenizer.perl
 - KenLM language model with binarization
 - word alignment with fast align + symmetrization
 - phrase extraction using Moses tools with binarization
 - evaluation (BLEU score) of a single sentence
- running the experiment on a cluster
 - single click to run the experiment on a configured cluster
 - the contents of intermediate output files can be seen in iEMS
 - status of each step in the training graph
 - possible to stdout/stderr of each tool
 - possible to stop and rerun each step
- modifying the experiment
 - change n-gram order for language model, only dependent steps are executed

- replace phrase extraction with phrase sampling by changing only one tool
- other areas of the interface
 - dashboard showing all experiments and cluster status
 - experiments area to filter & group experiments by language and other tags
 - cluster with an over of running hosts, queues and jobs
 - configuration properties of AWS, Vagrant and SSH hosts
 - launching AWS regular and spot instances from the interface
 - seeing the state, uptime, load average and other statistics about hosts
 - creating queues and assigning slots to them
 - terminating hosts

A working demo of the project was demonstrated at EAMT 2016 next to the poster about iEMS which is part of the product/project track.

In the project proposal, the following milestones were mentioned:

- Milestone 1: MT experiment designer
- Milestone 2: Multiple MT experiment management
- Milestone 3: Launch an MT environment on AWS with a single click

A lot of the functionality of all three milestones has been implemented.

Summary

Here is a summary of the new features.

- Created an interface for managing experiments.
- Major bug fixes, improvements and refactoring in the experiment designer.
- Machine translation tools now run in Linux containers.
- It is possible to launch new machines for running experiments with a single click:
 - Amazon Web Services EC2 regular instances
 - Amazon Web Services EC2 spot instances (which are heavily discounted)
 - VirtualBox (using Vagrant)
 - Connect to another machine using SSH
- Created a queueing system for scheduling machine translation jobs.
- Improved installation instructions.

Feature overview

Multiple sections

The very first prototype version of iEMS allowed to design just one experiment and there was no option to save changes and load them later.

Now, iEMS has several sections: dashboard, experiment manager, experiment designer, cluster view, queue management, host management and launch configurations.

The dashboard offers an overview of available experiments, running, pending and failed tasks as well as available machines for running experiments.

iEMS Experiments Cluster

Dashboard

Experiments

Name	Progress	Last modified
Phrase extraction with reordering	<div style="width: 100%; height: 10px; background-color: green;"></div>	2016-05-24 08:56
Phrase extraction	<div style="width: 100%; height: 10px; background-color: green;"></div>	2016-05-24 09:02
Phrase sampling	<div style="width: 100%; height: 10px; background-color: green;"></div>	2016-05-24 09:06

Jobs

Name	Slots	Jobs
Main	0 / 0	<div style="display: flex; gap: 5px;"> <div style="width: 10px; height: 10px; background-color: gray;"></div> <div style="width: 10px; height: 10px; background-color: yellow;"></div> <div style="width: 10px; height: 10px; background-color: green;"></div> <div style="width: 10px; height: 10px; background-color: red;"></div> </div>

Hosts

Launch Config	ID	State	SSH	Uptime	Load Avg.	Cores	RAM	Swap	Disk	Actions
AWS Micro	c-awsec2-31-190	running	connecting	00:00						Terminate
Vagrant Ubuntu 14.04	testconf9-459	running	connecting	00:00	0 0.03 0.08	1	226MB / 238MB	0B / 1024MB	227.2GB / 271.1GB	Terminate

Experiment management

In the experiment management section, it is possible to see all created statistical machine translation experiments. It is possible to group and filter them by user defined tags (such as languages, BLEU scores, etc.) as well as clone them with a single click to create an identical experiment but with different parameters.

It will also be possible to see how many steps of the experiment have completed with a progress bar (not implemented in the screenshot, just a placeholder for now).

iEMS Experiments Cluster

Experiments

Name	Progress	Last modified
Phrase extraction with reordering	<div style="width: 100%; height: 10px; background-color: green;"></div>	2016-05-24 08:56
Phrase extraction	<div style="width: 100%; height: 10px; background-color: green;"></div>	2016-05-24 09:02
Phrase sampling	<div style="width: 100%; height: 10px; background-color: green;"></div>	2016-05-24 09:06

[Create](#)

Search

Group by: -- None --

Source Language
[Select all](#) [Clear all](#)
 EN (3)

Target Language
[Select all](#) [Clear all](#)
 LV (3)

BLEU
[Select all](#) [Clear all](#)
 40.35 (1)
 40.84 (1)
 46.38 (1)

Experiment designer

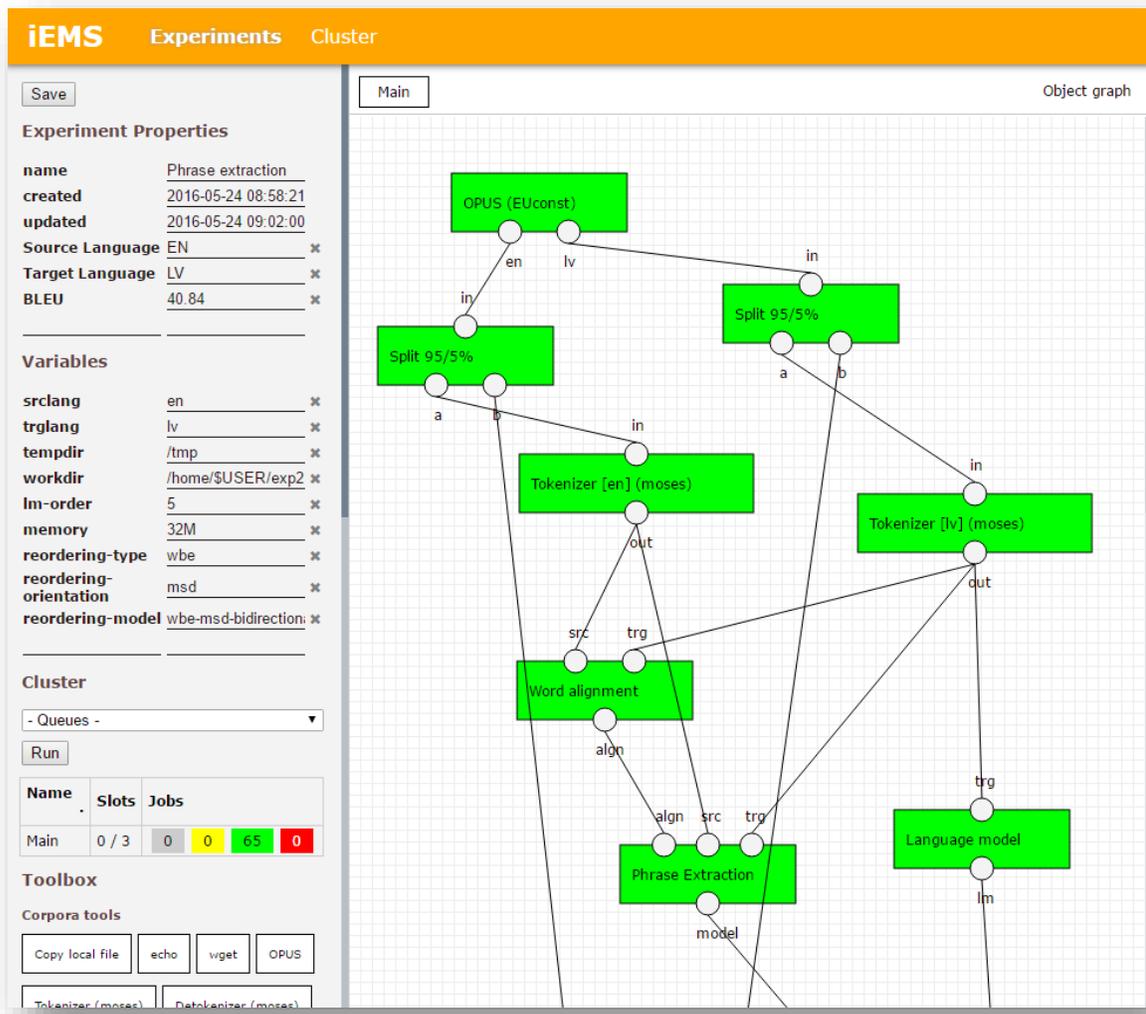
The improved experiment designer features a slightly better design but that is still being worked on.

Various bug fixes and a lot of refactoring has been done on the experiment designer.

It is now possible for the user to define their own properties for the experiment such as the source/target language, BLEU/TER score, possibly language model order, etc.

There is now parameter validation which means that it is immediately visible (there's a red border) that a process has not been configured properly. Parameter validation includes missing parameter values, invalid values (such as out of range values or string instead of an integer).

You can run experiments right from the designer in the cluster pane and also see the state of each step (pending, running, failed, stopped). It is also possible to see the output of (small) files by clicking on the input/output ports so that it is not necessary to connect to the machine to see the output of, say, the BLEU scoring script.



All machine translation tools that the users can drag & drop onto the design surface were refactored to add validation and rewritten to run in Linux containers (Docker).

There are presets of tools in the toolbox pane to run experiments out of the box with regular phrase extraction, regular phrase extraction with reordering, phrase sampling or CDEC using KenLM and fast align. Adding mGIZA/GIZA++, tuning and other tools is just a matter of hours.

Linux containers (Docker)

Linux containers are a great solution for distributing statistical machine translation software that includes all its dependencies (such as Perl and Python modules). I have included an explanation of what Linux containers are and why they are a great fit for SMT software at the end of this report.

During the development of iEMS, I created Dockerfiles that describe the creation of the Linux container images. I have uploaded the images to the public registry so that everyone can download and use them for free. All iEMS tools now run in Docker.

Repositories		Create Repository +	
Type to filter repositories by name			
iems/fastalign public	0 STARS	11 PULLS	> DETAILS
iems/kenlm public	0 STARS	10 PULLS	> DETAILS
iems/cdec public	0 STARS	10 PULLS	> DETAILS
iems/moses public	0 STARS	8 PULLS	> DETAILS
iems/opus public	0 STARS	3 PULLS	> DETAILS

Cluster

The cluster section allows the user to launch new machines on which to run experiments and also manage job queues.

iEMS Experiments Cluster

Cluster

Queues

Name	Slots	Jobs
Main	0 / 3	0 0 65 0

Hosts

Launch Config	ID	State	Uptime	Load Avg.	Cores	RAM	Swap	Disk	Actions
AWS EC2 Spot Micro 1GB/1vCPU	testconf1-704	running	00:34	0 0.01 0.05	1	569MB / 589MB	0B / 0B	1.5GB / 7.7GB	<input type="button" value="Terminate"/>
AWS EC2 Spot Micro 1GB/1vCPU	testconf1-697	running	00:34	0 0.01 0.05	1	498MB / 589MB	0B / 0B	1.2GB / 7.7GB	<input type="button" value="Terminate"/>
Vagrant Ubuntu 14.04 LTS x64	testconf9-890	running	74:42	0 0.01 0.05	1	189MB / 238MB	80MB / 4GB	239.1GB / 271.1GB	<input type="button" value="Terminate"/>

Vagrant Ubuntu 14.04 LTS x64 1 [Launch configurations](#)

It is now possible to launch new machines on the Amazon Web Services cloud. It is possible to launch regular machines or request spot instances. Spot instances are generally much cheaper (as low as 10% of the regular price) but can be terminated at any time without a prior warning.

It is also possible to launch a new virtual machine in VirtualBox (using a tool called Vagrant) which can be used for testing or for those who run Windows as their operating system on their powerful desktop.

Finally, it is possible to connect to an already running machine via SSH.

When a new machine is launched, it is provisioned, i.e., everything needed for running experiments is installed. Currently, the only requirement is to install Docker, and also add some SWAP.

In the cluster section, it is possible to monitor the uptime, load averages as well as used RAM and SWAP and disk space. Ideas for improvements include a historical chart of memory and CPU usage, maybe a file browser in the browser and a column that shows how much money has been spent on Amazon so far.

Launch Configurations

Amazon Web Services (AWS) Elastic Cloud Compute (EC2)

Name	Region	Instance type	Actions
AWS EC2 Spot Micro 1GB/1vCPU	eu-west-1b	t1.micro	Clone Edit Delete
AWS EC2 RRegular Micro 1GB/1vCPU	eu-west-1b	t1.micro	Clone Edit Delete

[Create](#)

Vagrant

Name	Image	Memory	Cores	Actions
Vagrant Ubuntu 14.04 LTS x64	ubuntu/trusty64	256	1	Clone Edit Delete

[Create](#)

Local SSH connections

Name	Host	Port	User	Actions
Linode		22		Clone Edit Delete

[Create](#)

Queues

Each experiment generates a bunch of tasks that I refer to as jobs. These jobs are then submitted to a queue. The user can assign one or more hosts to each queue and the jobs that are in the queue will be run on one of the hosts.

With the introduction of queues, it is now possible to run one experiment on several machines at the same time (more powerful machines can process more jobs simultaneously than a less powerful one) or you can create a queue for each experiment and assign a queue to run on a specific machine.

It is also possible to see the standard output/error logs as well as small output files right in the browser, so that it is not necessary to connect to the machine to debug what went wrong in case of an error.

Appendix: Linux containers (Docker)

First, I'd like to explain why Linux containers are necessary and a great fit for statistical machine translation software.

Let's say that you want to run Moses on your computer. One way to get Moses to run is to compile it from source. However, it may take up to half an hour to compile and even more time to resolve all problems with missing dependencies and libraries. For this reason, it is common to distribute compiled binaries for users. There may still be problems with missing libraries if the binaries were compiled with dynamic linking, so Moses by default sacrifices a little bit of disk space and compiles all binaries statically to avoid such problems.

Unfortunately, static linking can't solve all issues. Moses is not just a decoder, it is a suite of tools (usually written in Perl and Python) that require other modules to be installed.

One way to get rid of all these problems is to just set up everything as needed in a virtual machine and offer it as an image to others. For instance, this can be done in VirtualBox and the resulting disk image can be given to others. Amazon Web Services have their own image format that, for instance, the Box¹ project uses.

There are several problems with disk images. First, virtualization has a lot of overhead, the disk images may be large and the way the disk image was set up may not be known or reproduced. The images on Amazon Web Services can only be run on the Amazon cloud.

Another solution is Linux containers. Linux containers allow you to run programs as well as other Linux distributions on an already running Linux system. The containers are isolated and have almost no overhead (no virtualization is used). If you are not familiar with containers, you can think of them as chroot on steroids, except you should not expect containers to be secure.

Docker is a very popular container solution on Linux. It is possible to script and version the creation of containers or you can think of it as VirtualBox + Git version control.

The advantages of containers for statistical machine translation software are numerous. Since decoders and other processes are mostly CPU-bound, they have better performance if no virtualization is used. It is no longer necessary to spend tens of minutes compiling software from source or installing missing dependencies or other libraries.

The first time a tool that is run in Docker is used, it is downloaded from the registry (generally, the container size is the sum of the size of all binaries, if there are Perl dependencies, a base image of Ubuntu Server is used which adds extra 100 MB).

¹ <http://www.boxreasear.ch/index.html>