Lab 4: IBM1 / get familiar with NLP-Unix

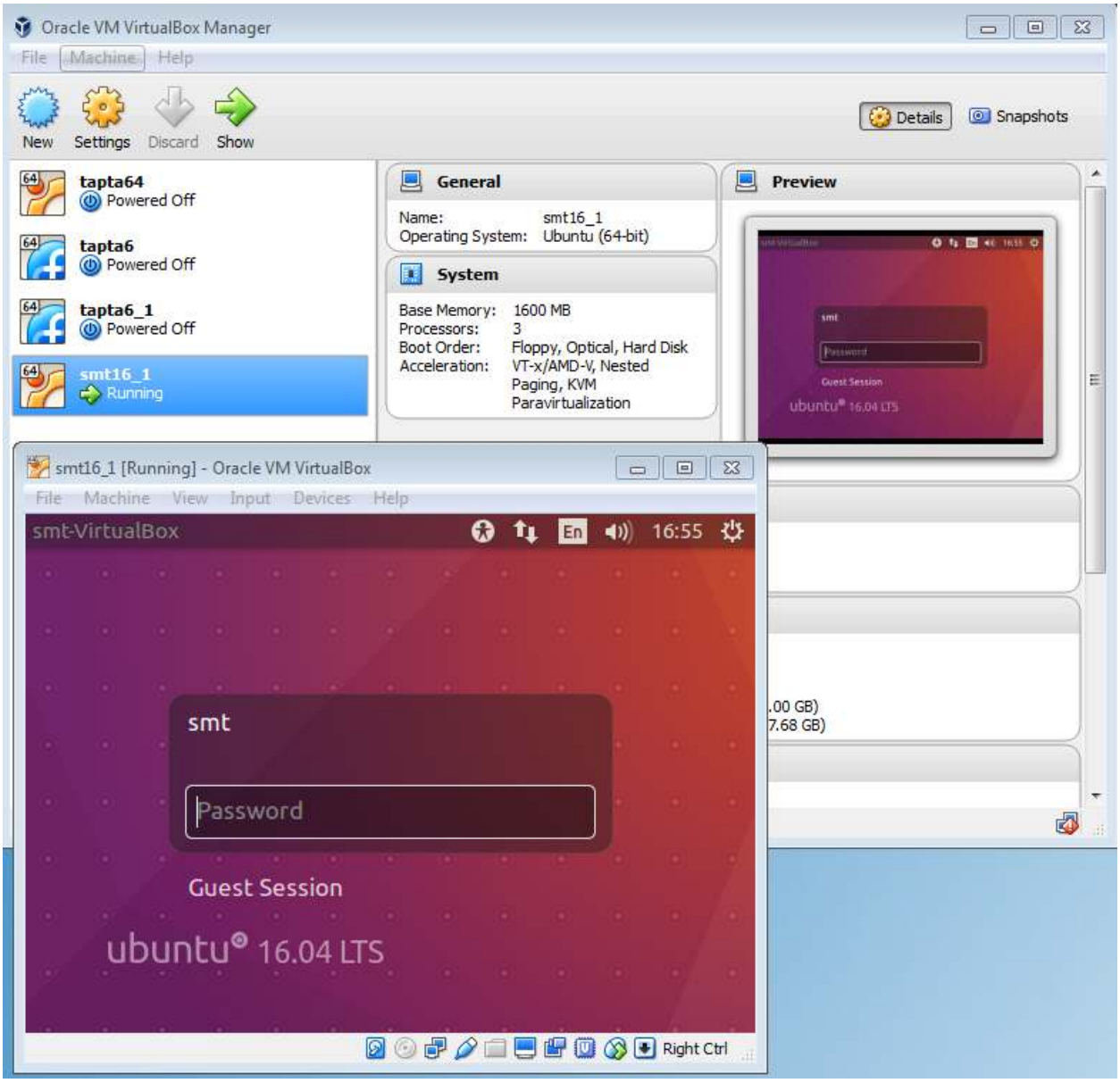
This documentation contains instructions to follow the lab n.4 GIAN course about MT .

Status: 12/12/2016 BP V0.03

Contents

Preliminaries

1. Your host compute contains a “SMT” virtual machine provided
2. If not, please open virtual box (on Ubuntu, click on “search”, type virtual, open it)
3. In File, choose “import appliance”, select the file name (smt…ova)
4. The virtual machine will appear, select the machine, click “start”



The username is “smt” and password is the same

Install the necessary packages

Open a terminal and type:

```
cd Icon2016
git up
```

(if the previous command does not work, type:

```
cd
git clone https://github.com/mlearningbruno/giancourse.git Icon2016
```

it should display something like:

```
Cloning into 'Icon2016'...
remote: Counting objects: 27, done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 27 (delta 4), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (27/27), done.
Checking connectivity... done.
```

You can now launch the "lab4 install script":

```
bash Icon2016/labs/lab4install.sh
```

➔it might take few minutes

Choose your assignment: Get familiar with Unix / develop IBM model 1

1. If you are a developer: choose IBM1
2. If you are not: prefer getting familiar with Unix

IBM Model 1:

If you understood the last lecture (if not, please read again the slides: Lecture4.3-LearningLexicalTranslations.pdf), develop in your favorite language the following pseudo code:

Input: set of sentence pairs $(\mathbf{e}, \mathbf{f})$
Output: translation prob. $t(e|f)$

initialize $t(e|f)$ uniformly
while not converged **do**
  // initialize
  count$(e|f) = 0$ **for all** e, f
  total$(f) = 0$ **for all** f
  for all sentence pairs (**e**,**f**) **do**
    // compute normalization
    for all words e in **e** **do**
      s-total$(e) = 0$
      for all words f in **f** **do**
        s-total(e) += $t(e|f)$
      end for
    end for
    // collect counts
    for all words e in **e** **do**
      for all words f in **f** **do**
        count$(e|f)$ += $\frac{t(e|f)}{\text{s-total}(e)}$
        total(f) += $\frac{t(e|f)}{\text{s-total}(e)}$
      end for
    end for
  end for
  // estimate probabilities
  for all foreign words f **do**
    for all English words e **do**
      $t(e|f) = \frac{\text{count}(e|f)}{\text{total}(f)}$
    end for
  end for
end while

Do not look for efficiency for now: simply make it run with some easy example sentences, then try to read the sentences from two tokenized texts:

parallel-corpora/hi-en/testset.en parallel-corpora/hi-en/testset.hi (same for other languages, change “hi” by “bn”...”ur”)

Once you can make it run on this test (1000 lines), try it with a bigger corpus (here the trainset ~27000 sentences):

parallel-corpora/hi-en/testset.en parallel-corpora/hi-en/testset.hi

Getting familiar with NLP and Unix commands

Please read the attached documentation "UnixCommandsInANutshell"

Then, try to do the following:

1. Open a terminal
2. Unpack the "parallel corpus"
3. Go into the "hi-en" directory
4. Count the number of lines in the file "trainset.hi"
5. Count the number of lines of all the files trainset, devset, testset
6. Count the number of words in English files (trainset, devset, testset)
7. Find a way to count the number of non-duplicated lines in the trainset (for English and for Hindi), observe the difference and relate it to the original corpus (same Hindi sentence has been translated several times)
8. Look for sentences in the English trainset that contain "world"
   Too many lines are displayed

9. (same as previous) but using more to look at results page by page
10. Look at testset putting side-by-side English and Hindi sentence
11. Find a way to automatically display Hindi sentence side-by-side with English sentence that contains the word "worlds"
    a. Sort previous result by alphabetical order
    b. Display only the number of lines

Launch a Moses training on one language pair (e.g. hi en)

Use the provided script:

bash Icon2016/labs/trainMoses.sh hi en

⇨ It will create a full MT model that you can now use with the following command:
/home/smt/mosesdecoder/bin/moses -f parallel-corpora/hi-en/trainMoses/moses.ini

1. Use previous command to directly translate a given sentence,

e.g.

```
echo "मैं दुकान जाएगा" | mosesdecoder/bin/moses -f parallel-corpora/hi-en/trainMoses/models/moses.ini
```

2. Use previous command to translate the Hindi testset

3. Save the result in a file

   a. Check that the input file and the output file have exactly the same number of lines
   b. Display them side-by side (original and machine translation)