Moses for Mere Mortals

Tutorial

A Machine Translation chain for the real world

Maria José Machado
Hilário Leal Fontes

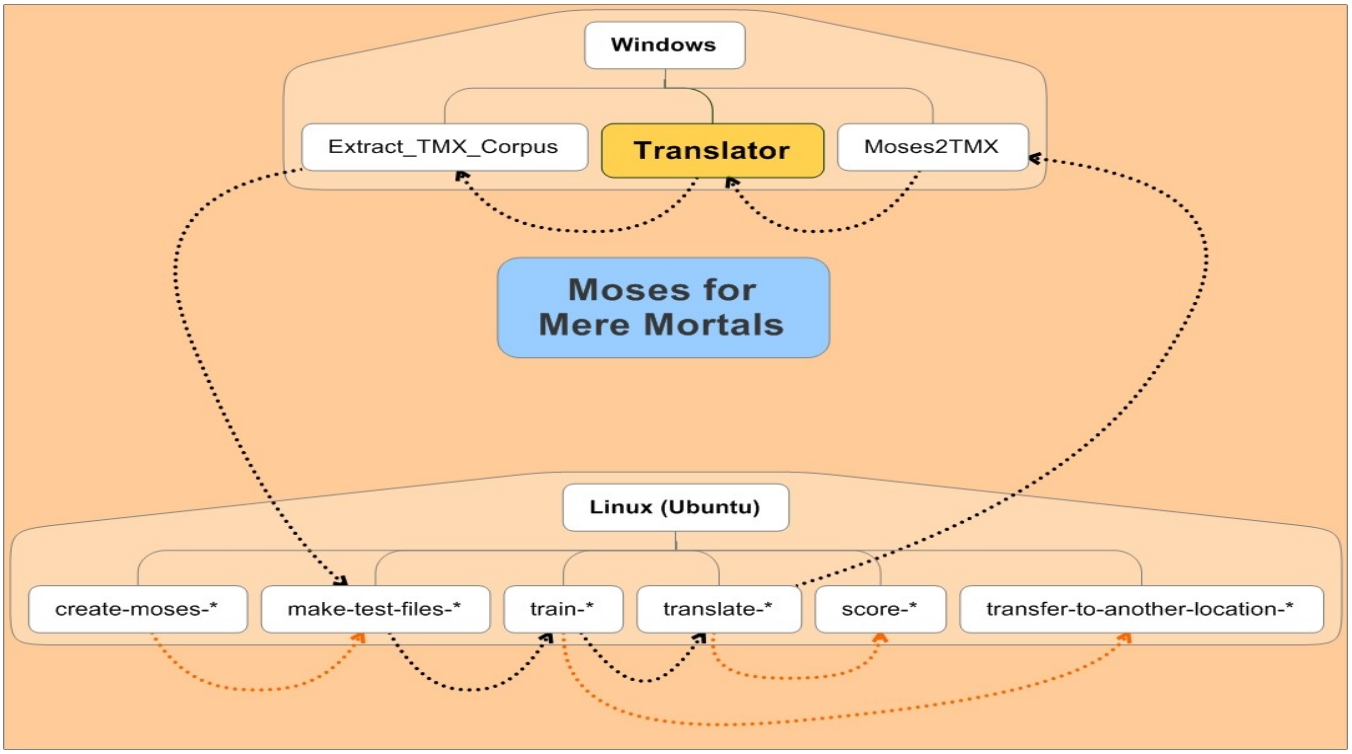
November 2014

Foreword

Moses for Mere Mortals (MMM) is a machine translation package intended to be user-friendly and understandable by users with no background in computational linguistics or computer science. This is its leitmotiv and that is why it is called 'for mere mortals'.

MMM makes available to a wider number of users the world-renowned and state-of-the-art open-source Statistical Machine Translation System ***Moses*** *that is used in research and in commercial and public MT applications all over the world.*

With MMM you can create a real-world machine translation chain — for almost any language pair — with small or large corpora (of even more than 10 million segments) in a domestic PC/laptop in a few hours or days, depending on the size of your corpus and on your computer capacity.



Bear in mind that in order to get the best results, the corpora used for training should contain many — and preferably all — of your personal (or group) previously translated documents (translation memories) relating to the domain to be translated. Of course the more (pertinent) data you have — and of good quality — the better.

Furthermore, MMM scripts — which were first developed by a translator for translators — also enable an easy integration of Moses in the translators' workflow.

It includes a set of addins so that you can — in batch operations: i) convert (your) TMX files into ready-for-Moses corpora, and, ii) afterwards convert Moses translations into TMX files that you can use as a translation memory in open-source or commercial CAT tools (like OmegaT or MemoQ).

So, the more you translate, the more material you will then have to use in your own (or your group's) machine translation system.

The only requirements needed to use MMM are to have some basic knowledge of the Linux operating system and an Ubuntu distribution — 12.04 (LTS) or 14.04 (LTS) (64 bits) — in your computer.

Since installing and using such a sophisticated MT system is probably a daunting prospect that few non-experts will dare to attempt unguided, the Section on ***What Makes Moses Tick*** *(Annex 1) presents an (outrageously!) simplified overview of basic concepts about Statistical Machine Translation in general — and Moses in particular — to give absolute beginners an idea of what Moses is about.*

Also, in the Section ***Further reading and videos****, we have selected — from the large amount of information available on the Internet — some information that you may find interesting.*

This may help you to better understand the instructions in this Tutorial... and/or to understand why sometimes the MT output is so astonishingly good and other times pure rubbish... and watch out for pitfalls.

With MMM, you will be able to install both Moses and the packages upon which it depends with a single command and the same will happen (just one command) when you train your corpora and then translate your documents, having previously defined some settings in the script files (which are described in this Tutorial with comments to guide you).

MMM has a Demo with a 200000 segment corpus — too small to do justice to the qualitative results that can be achieved with Moses — that will allow you to quickly see how it works and check that MMM is working correctly in your computer.

As the EU 50 years' policy of multilingualism has produced a large body of high quality multilingual corpora in all the EU official languages, there is now a substantial amount of data that is freely available on the Internet and which you can use to build your own MT engines.

As translators are — after all — those who provide the raw materials in the form of high-quality human translations that make SMT possible, it is only fair that they have the possibility to leverage their work with computational means as affordable as a 1000 Euro computer … and an acceptable amount of time and effort!

Although MMM is meant for users that are not experts in Machine Translation, the ***train*** *and* ***translate*** *scripts give you the possibility of 'playing around' with different settings as you can easily define 70 parameters in the* ***train*** *script and 18 parameters in the* ***translate*** *script.*

We know that MMM is now being used by students in universities and that it has also been integrated in the "complete suite of professional CAT tools" developed by the Swiss-based Olanto Foundation[1].

We sincerely hope that you will find this updated version of Moses for Mere Mortals useful.

[1] http://olanto.org/software/mymt/documentation)(http://olanto.org/docs/Installing_Back-End_Server_V2.0.pdf

Table of Contents

(SEE DETAILED INDEX AT THE END OF THIS TUTORIAL)

I — Overview

Moses for Mere Mortals' main objectives are:

A. To **guide the first steps of users** who are just beginning to use Moses, including by giving an overview of basic concepts in Statistical Machine Translation in general — and Moses in particular — in Annex 1 on **What Makes Moses Tick**;

B. To **quickly test the installation of Moses with a** PT-EN **demonstration corpus** of 200 000 segments (about 3.8 million words) after having downloaded and installed MMM, just by running, with a single command, each of the several scripts (see Part 1).

C. To help build a **machine translation chain for the real world** (with your own corpora, your own settings and your own documents to translate) so that you can easily:

1) Install all the Ubuntu packages needed to be able to compile both Moses and Moses for Mere Mortals — using the ***install*** script;

2) Install Moses — the decoder — and all the components necessary for it to work — word aligner, language model(s) and also scorer(s) — with a single instruction — using the ***create*** script;

3) Prepare the corpora to be trained with Moses, i.e. corpora from translation memories (either public or your own TMXs) — using the ***EXTRACT_TMX_CORPUS_1.043.EXE***.

   Take into consideration that some languages, like Chinese or Arabic, might require special corpora pre-processing tools that are not provided in MMM;

4) Generate test and tuning files from the corpus to be trained by extracting from it 2 test files and 2 tuning files with randomly selected, non-consecutive segments that are erased from the training corpus files — using the ***make-test-files*** script;

5) Train corpora to build MT engines — using the ***train*** script;

6) Translate documents — using the ***translate*** script;

7) Convert the source and MT target files into TMX format to be used in CAT tools — using ***MOSES2TMX_1.032.EXE***;

8) Quickly evaluate MT output quality with automatic metrics — BLEU and NIST scorers — for one document or for a batch of documents placed in a single directory — using the ***score*** script;

9) Transfer the trainings in a MMM installation to another location in the same computer or to another computer — ***using the transfer-to-another-location*** script.

II — Scope and some interesting features

The *Moses for Mere Mortals* scripts make it easy to use Moses to train **phrase-based machine translation** engines where correspondences are simply between continuous sequences of words. This kind of training already gives useful results for a substantial number of language pairs and is the basis of systems like Google and Bing.

With Moses, it is also possible to train corpora where every word is presented together with, for instance, its respective lemma and/or part of speech tag (**factored training**), or where more structure is added to the correspondences as happens with **hierarchical phrase-based** or **syntax-based** machine translation.

This is particularly important for language pairs in which the target language is morphologically-rich. However it substantially increases the complexity, the computer capacity and the time needed to train the corpora and these are very active areas of research at the moment.

The present *Moses for Mere Mortals* scripts don't cover these types of training. But even so, MMM allows you to select 70 settings for **phrase-based training** which represent an enormous amount of possible combinations, some of which you may want to try to improve the quality of MT output for your particular language pair, corpora and translations.

However, don't worry. You don't have to change the huge majority of settings if you don't want to go that deep!

Just by reading Parts 1 and 2 of this Tutorial, you will be able to train a corpus and translate your documents as MMM has defaults defined for each parameter. You can just accept them and — even so — produce Moses translations that might be useful to you.

MMM also has some interesting features, namely:

1) In terms of corpora preparation, control characters are automatically removed from the input files as these can crash a training;
2) Selected parameters of the Moses scripts, the Moses decoder and of the other packages can be controlled in the script files;
3) The training is stopped with an informative message if any of the phases of training (language model building, recaser training, corpus training, memory-mapping, tuning or training test) doesn't produce the expected results;
4) Tuning can be done separately with reuse of a previously training without tuning;
5) The duration of tuning can be limited (tuning is a very time-expensive phase; the present version of MMM includes 2 new tuners, pro and kbmira);
6) Binarisation (Memmapping) of the phrase table and reordering table are included, which allows to translate with a reasonable amount of RAM;
7) Transferring your trainings to another computer or to another Moses installation in the same computer is easy.

III — Further reading and videos

There is a lot of information on the Internet both on Machine Translation/Moses and on Linux/Ubuntu. Here are some selected references.

III.1. Information on Linux/Ubuntu

- For information on Linux in general: The Story of Linux[2], Linux Foundation, 2011
- For information on Ubuntu 14.04 (latest Long Term Support Version): Getting started with Ubuntu 14.04[3], 2014
- For information on installing Ubuntu in a Virtual Box: Tutorial — Installing Ubuntu 14.04[4] J. LaCroix, 2014

III.2. Information on Moses and Statistical Machine Translation

- *How do Computers Learn a New Language? -- An Introduction to Statistical Machine Translation*[5], META-NET, 2013
- *Principles of Machine Translation*[6], Barry Haddow, TAUS MosesCore On Line Tutorial on Machine Translation and Moses, 2014
- For information specifically on Moses, go to the Moses website[7] and see particularly the *Moses User Manual and Code Guide*[8].
- Further information on Moses is also available in the MosesCore research project website[9].
- For a history of Machine Translation and lots of material, see John Hutchins website[10].
- For information on the general concepts of Statistical Machine Translation, read the book '*Statistical Machine Translation'* by Philipp Koehn[11], Cambridge University Press, 2010.
- For information on Machine Translation, see also the TAUS website and particularly the sections on Machine Translation[12] and Moses free tutorials[13].

2 https://www.youtube.com/watch?v=5ocq6_3-nEw
3 https://www.youtube.com/watch?v=SGGj3hc6R8Q&list=PLsMLAwgiqdxl5U6m9ZnefcfC0xncvLCL5
4 https://www.youtube.com/watch?v=i_4Kh5kE3xA
5 https://www.youtube.com/watch?v=_ghMKb6iDMM
6 https://www.youtube.com/watch?v=beX5rqdneII
7 http://www.statmt.org/moses/
8 http://www.statmt.org/moses/manual/manual.pdf
9 http://www.statmt.org/mosescore/
10 http://www.hutchinsweb.me.uk/
11 http://homepages.inf.ed.ac.uk/pkoehn/
12 https://labs.taus.net/mt

- For a point of view about MT in Translation Studies, see '*Revising Machine Translation: A Marketable Skill*'[14], by Belinda Maia, University of Oporto, Presentation at the Eramus Network for Professional Translator Training, 2013.
- For another point of view about MT in Translation Studies, see '*Statistical machine translation in the translation curriculum: overcoming obstacles and empowering translators*'[15], by Dorothy Kenny and Stephen Doherty, *in The Interpreter and Translator Trainer*, vol. 8, issue 2, 2014
- For information on the Machine Translation market and open-source and commercial systems available, see the '*Machine Translation Market Report*'[16], by Jaap van der Meer, Achim Ruopp, TAUS, August 2014.
- For a review on ongoing MT research, see *Open Problems in Machine Translation*[17], Philipp Koehn, Inaugural Lecture, University of Edinburgh, 2013 .
- For information on how Google works, see *Statistical Machine Translation — Breaking down the language barrier*[18], Franz Josef Och, Google Faculty Summit 2009: Statistical Machine Translation, GoogleTechTalks, 2009
- For more information on MT, see Popular Machine Translation Videos[19], YouTube
- For a point of view on open-source software as applied to translation: *Free and open-source software - A translator's good friend*[20], Maria José Bellino Machado, *a Folha*, no. 45, Summer 2014.
- For some information on corpora for Portuguese: *European Union multilingual corpora for use in translation*[21], Hilário Leal Fontes, *a Folha*, no. 45, Summer 2014.

13 https://labs.taus.net/mt/mosestutorial
14 http://www.translator-training.eu/attachments/article/65/Revising%20MT.ppt
15 http://www.tandfonline.com/doi/abs/10.1080/1750399X.2014.936112?journalCode=ritt20
16 https://www.taus.net/mt-market-report-2014
17 https://www.youtube.com/watch?v=6UVgFjJeFGY
18 https://www.youtube.com/watch?v=y_PzPDRPwlA
19 https://www.youtube.com/watch?v=_ghMKb6iDMM&list=PLb8-znkGDvF2O2nVwO8oZZF635YcHDpll
20 http://ec.europa.eu/translation/portuguese/magazine/documents/folha45_foss_en.pdf
21 http://ec.europa.eu/translation/portuguese/magazine/documents/folha45_corpora_en.pdf

IV — Authorship and collaborations

Moses for Mere Mortals, EXTRACT_TMX_CORPUS_1.043.EXE and MOSES2TMX_1.032.EXE: João Luís Rosas (moses.for.mere.mortals@gmail.com)

Corpora compilation and cleaning: Hilário Leal Fontes

Testers: Maria José Machado (mj.bellino.machado@gmail.com) and Hilário Leal Fontes, who tested the scripts and contributed with suggestions and feedback in a real-life translation workflow.

Tutorial: Maria José Machado, João Luís Rosas and Hilário Leal Fontes

V — Thanks

Special thanks to:

- **Maria José Machado,** who helped in the evaluation of Moses output in general and organised, together with Hilário, a comparative evaluation, made by professional translators, of the qualitative results of Google, Moses and a rule-based MT engine.
- **Hilário Leal Fontes**, who made helpful suggestions and tests. He is the author of the non-breaking_prefixes.pt file for the Portuguese language. He compiled the corpora that were used to train Moses and to test these scripts, including 2 very large corpora with 6.6 and 12 million segments, besides the 3.4M segment corpus used in examples in this Tutorial.
- **Sérgio Portugal,** who designed the Moses for Mere Mortals logo (© MMM).
- **Manuel Tomas Carrasco Benitez**, whose Xdossier was used to create a package for the Moses-for-Mere-Mortals files.
- **Authors of the http://www.dlsi.ua.es/~mlf/fosmt-moses.html (Mikel Forcada and Francis Tyers) and of the http://www.statmt.org/moses_steps.html pages**. These pages helped a lot in first steps with Moses.
- **Authors of the documentation of Moses, MGIZA, IRSTLM and RandLM;** some of the commentaries of the present scripts describing the various settings include citations of them.
- **European Commission's Joint Research Center and Directorate-General for Translation** for the DGT-TM — freely available on the JRC website and providing aligned corpora of Community law texts in 22 languages - which was used for the tests.
- **Kelde Pulo,** who helped in the optimisation of the Linux installation in order to get the best performance out of some of our computers.
- **Tom Hoar,** who consolidated the previous documentation into the very first version of the Quick-Start-Guide.doc to help users to get up to speed very quickly.
- **Gary Daine**, who made helpful remarks and who contributed code for Extract_TMX_Corpus, the predecessor of EXTRACT_TMX_CORPUS_1.043.EXE.

VI — Licence

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation (version 3 of the License).

This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details (http://www.gnu.org/licenses/).

VII — Symbols used in this Tutorial

Throughout this Tutorial we call your attention to:

 Important details

 Important risks

 Some tips based on our experience that might be useful to you

 Videos

Throughout this Tutorial — and to give you an idea of what is involved — we give you some indications concerning:

 RAM / swap memory

 Processors

 Disk space.

 Time to complete an operation

 Level of automatic scores to be expected (BLEU scores) for the PT-EN language pair

VIII — Corpora used to test MMM

We have tested the present release of Moses for Mere Mortals mainly with PT-EN and PT-FR corpora of up to 12 million segments.

The following PT-EN corpora were extracted from the European Commission's Joint Research Centre website and are used as examples in this Tutorial:

Demo: The Demo corpus (with about 200 000 segments — 3.7 million words (EN))[22].

Corpus-3.4M: A 3.4 million segment corpus (with 64.4 million words (EN)). This corpus has been optimised and converted to the New Portuguese Spelling[23].

IX — Computers used in the examples

The computers used for the testing examples mentioned in this Tutorial were:

PC1: Desktop PC with CPU Intel Quad Q8300 (4 cores), 8 GB RAM, 2 hard disks (2 TB each) and a 80 GB SSD disk (for swap) (2010)

PC2: Desktop PC with CPU Intel i7 3770 (4 cores, 8 threads), 32 GB RAM, 4x250 GB SSD disks in RAID for trainings and 2 internal hard disks (2 TB each).

[22] DGT Translation Memory 2007: http://optima.jrc.it/Acquis/DGT_TU_1.0/data/

[23] DGT-Translation Memory 2007-2013: https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory

PART 1
– INSTALLING MMM AND RUNNING THE DEMO –

1 — Installing Moses for Mere Mortals

So as to avoid having to update this Tutorial every time a script version is changed, the version numbers of the scripts have been omitted (for example, in most places we write ***create-**** (or just ***create***), instead of ***create-1.43***, in order to refer to the ***create-1.43*** script).

The first time you install MMM, you must previously run the ***install*** script in order to automatically install all the Ubuntu packages needed to compile Moses and Moses for Mere Mortals.

For Moses — the Decoder — to work, it needs several other applications which have been developed by other researchers/organizations.

MMM installs all the needed packages with a single command so that in a few minutes — depending on the speed of your Internet connection — you can start using Moses right away.

1.1. Requirements

1.1.1. System requirements

Moses for Mere Mortals (MMM) has been tested with the following Linux (64 bits) distributions, which are Long Term Distributions:

- Ubuntu 14.04 LTS (http://www.ubuntu.com/download/desktop) — long-term support release until April 2019.
- Ubuntu 12.04 LTS — long-term support release until April 2017.

MMM should also work in other Linux distributions with slight changes, but no such tests have been carried out.

You can install Ubuntu in your computer as its only operating system or with dual booting if you want to use another operating system like Windows in the same computer.

Furthermore, you can also install Ubuntu in a Virtual Box but only if you just want to test Moses for Mere Mortals with small corpora as Moses requires some computational capacity and disk space to train medium/large corpora.

MMM was tested in a Virtual Box machine with Ubuntu 12.04. It installs all the packages needed for all the software that it uses.

1.1.2. Minimum computer requirements:

4 GB of Random Access Memory, but preferably much more (we would recommend, at least, 8 GB if you want to train medium/large corpora)

2 processors, but preferably a fast multiprocessor computer (we would recommend an 8 cores/threads if you want to train medium/large corpora)

1 hard disk (at least 0.5 TB), depending on the size of the corpora you want to train.

As a rule of thumb, the disk space needed for training is approximately 100 times the size of the training corpus (source plus target files).

Just as an example — and for you to have an idea of what you are getting into — to train a 3.4 million segment corpus, with about 1 GB for the EN and PT training corpus that we are using as an example in this Tutorial:

You will need 100 GB free disk space just to train it so that it has space to manage all the temporary files created in the training process. The size of the final training will be about 11 GB.

This corpus was trained (with the default settings) in about 40 hours in PC1 and in about 22 hours in PC2.

The BLEU score was 78.44 using the test corpus generated automatically by MMM from the original corpus. However, take into consideration that, as the testing corpus was extracted from the base corpus and is therefore an in-domain testing corpus, the scores for entirely new documents can be (much) lower!

1.1.3. Software requirements

MMM installs all the necessary packages.

1.2. Requirements to install Moses for Mere Mortals in Ubuntu

To install Moses for Mere Mortals you must have administrator rights in the computer you are using (if you install Ubuntu with its defaults, then you have them).

You can copy the uncompressed MMM to any location you want in your computer but we recommend that you start by copying it to a **Machine-Translation** folder that you previously create under **$Home/Desktop**, at least for testing purposes, as this is the default in all the MMM scripts. However, you can change it if you want.

MMM includes a **Demo** which we also highly recommend you to run so that you can have an idea of how MMM works and also check that it is working correctly in your computer.

MMM contains all the MT packages necessary for you to create a MMM installation without the need for an Internet connection.

However, for Moses to work, there are some Linux packages that probably need to be installed in your computer and for that you do need an Internet connection. So, you must run the ***install*** script before creating a MMM installation for the first time in a computer.

If afterwards you want to create another MMM installation (you can have as many as you want, for different language pairs, for instance), you can do it without being connected to the Internet — and without running the ***install*** script — as all Linux packages have already been installed in that particular computer.

1.3. Preparation to install MMM in Ubuntu

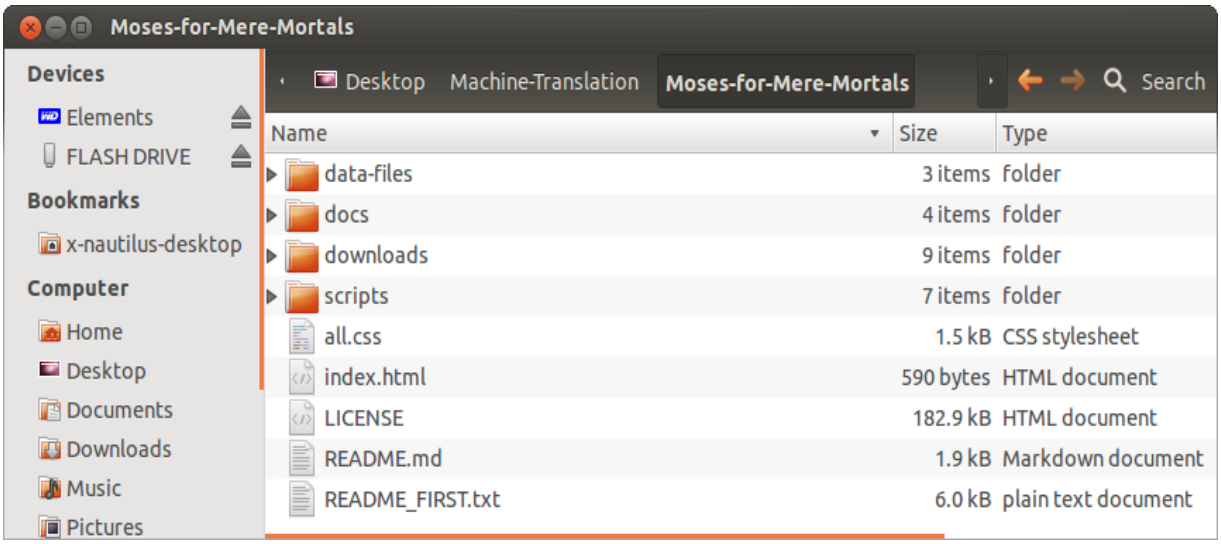
So as to avoid having to update this Tutorial every time a script version is changed, the version numbers of the scripts have been omitted (for example, in most places we write ***create-**** (or just ***create***), instead of ***create-1.43*****,** in order to refer to the ***create-1.43*** script).

1) In Ubuntu, access the https://github.com/jladcr/Moses-for-Mere-Mortals/releases page
2) Click on the "**Source code (tar.gz)**" button just under the Moses-for-Mere-Mortals-1.23 release.

   A dialog window will open with a "Save File" option.
3) Select it and click the 'OK" button.

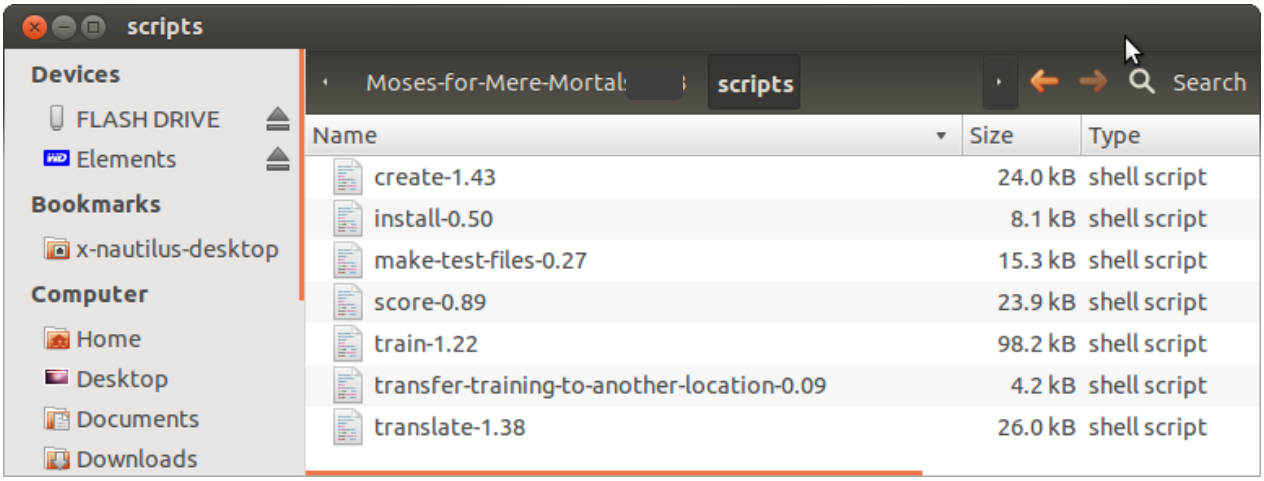
   The full download will then start.
4) Uncompress the downloaded release (tar.gz), e.g. by right clicking on the name of the file and selecting ***Extract Here***. You will see a ***Moses-for-Mere-Mortals**** folder;
5) In the **File Manager**, create a folder under ***$HOME/Desktop*** called **Machine-Translation**;
6) Copy the ***Moses-for-Mere-Mortals-**** uncompressed folder to the ***$HOME/Desktop/Machine-Translation*** folder you created;
7) Rename it ***$HOME/Desktop/Machine-Translation/Moses-for-Mere-Mortals***

8) Double click on the Moses-for-Mere-Mortals folder to see the subfolders it contains:



Screenshot 1 — *Moses-for-Mere-Mortals* folder

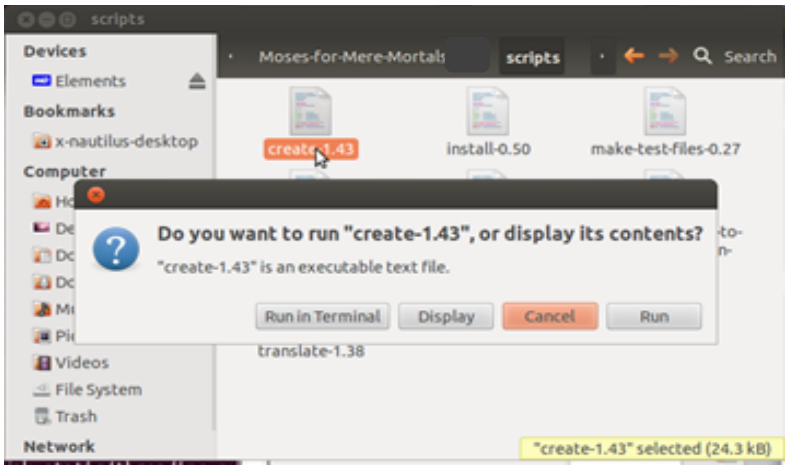
9) Double click on the ***scripts*** folder and see the files it contains.



Screenshot 2 — *scripts* folder

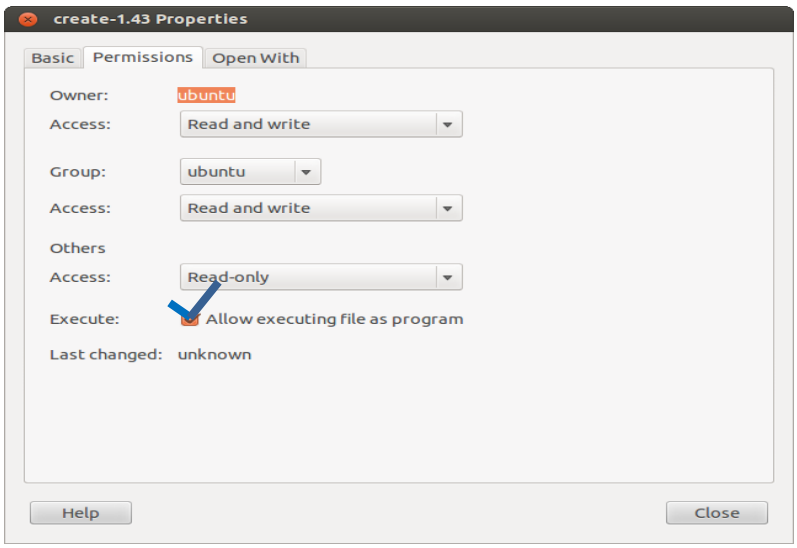
10) It may happen that, when you copy the MMM package, the executable scripts files it contains loose the executable attribute.

To check that the 7 ***script*** files in this folder named ***install-****, ***create-****, ***make-test-files-****, ***train-****, ***translate-****, ***score-**** and ***transfer-training-to-another-location-**** are executable, double click on the name of one of the script files. If you see the 4 options below (***Run in Terminal***, ***Display***, ***Cancel*** and ***Run***), it is all right and you can skip the next points.



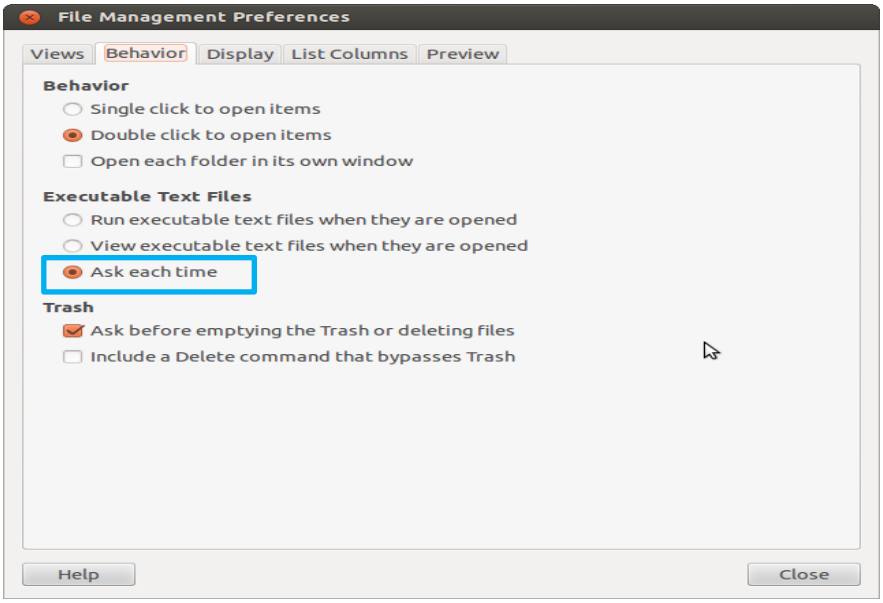
Screenshot 3 — The *scripts* folder with the executable script files

11) If not, right click on the name of the first script and in ***Properties*** — ***Permissions***, click on ***Allow executing file as program***. Repeat the operation for all the scripts.



Screenshot 4 — Making the script files executable

12) If it still doesn't show the 4 options, check in Ubuntu — in **Edit** — **Preferences** — **Behaviour**, under **Executable text files**, to see if the box **Ask each time** is ticked. If not tick it.



Screenshot 5 — *File Management Preferences* in Ubuntu to make the scripts executable.

1.4. Creating a Moses-for-Mere-Mortals installation

1) In your first installation of MMM in a particular computer, open the **Terminal** and enter:

   cd $HOME/Desktop/Machine-Translation/Moses-for-Mere-Mortals/scripts.

   See Section 3.12 if you have never used the **Terminal** before.

2) Enter ***./install-*.***

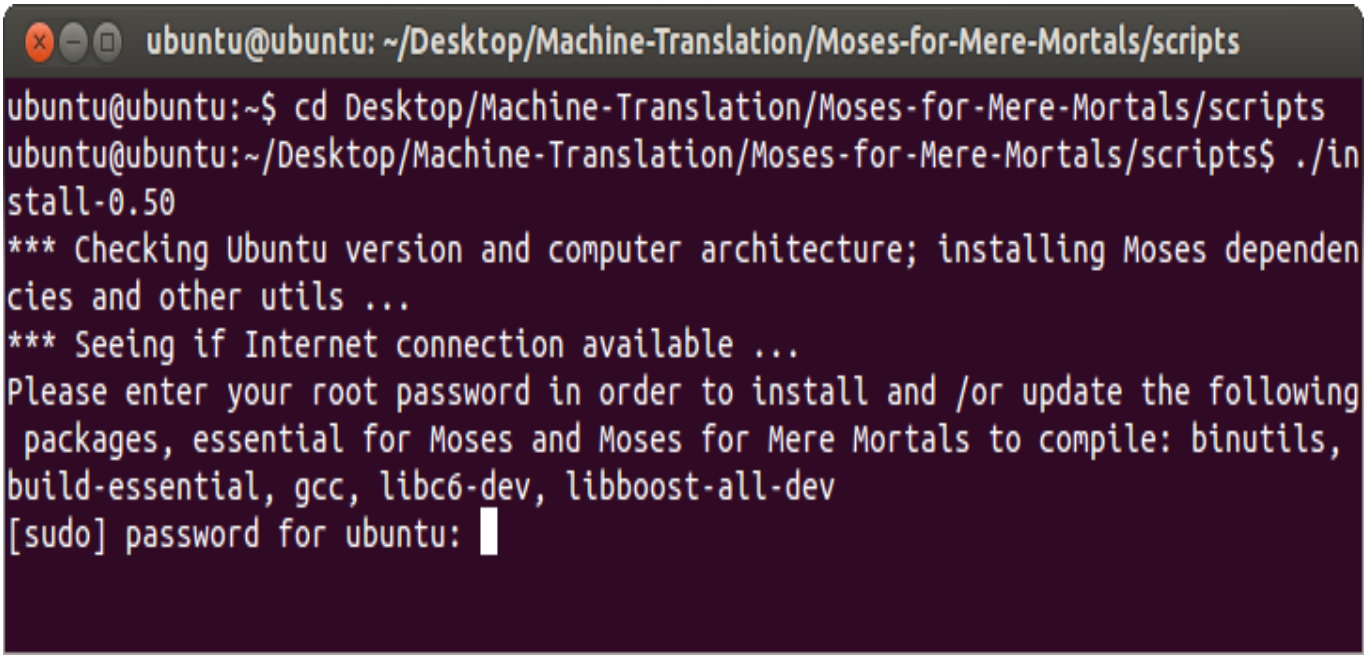
   You will be requested your administrator password (which is normally the password you use when you log on to Ubuntu).

   You can also double click on the ***install*** script and select ***Run in Terminal***, but — if there are problems — you will not see the error messages as the **Terminal** window will close as soon as the operation ends.

   However, in the ***Machine-Translation/MMM/logs/*** folder you have subfolders with the logs of each operation that runs on the **Terminal**. So, if you prefer not to use the **Terminal**, you can always check if everything was fine in the respective log.

3) Enter your password. The password is not displayed when you type it.

MMM will automatically install all the Ubuntu packages necessary to install Moses/MMM. You only need to run this script once in a given computer.



Screenshot 6 — Installing Ubuntu packages

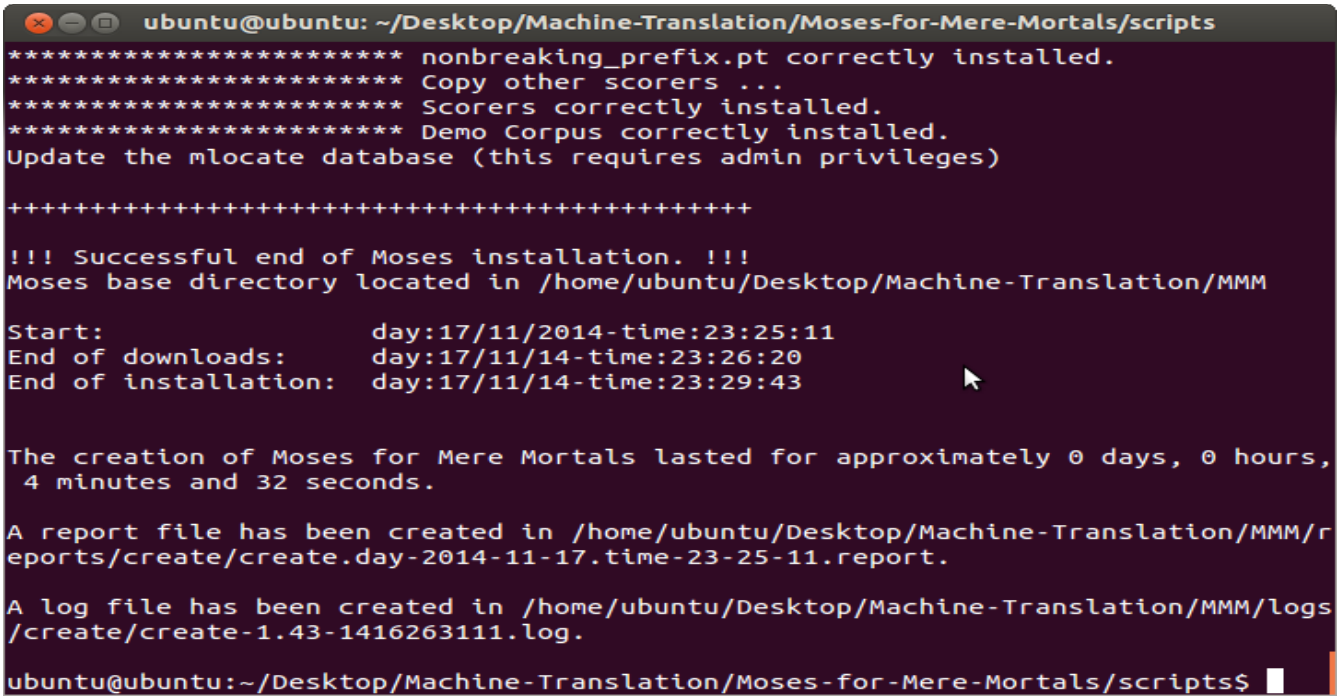
At the end, you will see a message indicating that all the packages were successfully installed.

4) In the **Terminal**, enter ***./create-**** to create a MMM installation.

MMM will request you administrator password.

5) Enter your password and MMM will install all the packages needed for Moses to work without any other intervention from you. It will take some minutes.

6) When the script ends, in the **Terminal** window you will — hopefully — see the information ***! ! ! Successful end of Moses Installation. ! ! !,*** followed by the name of the folder where Moses was installed, the time it took to complete the installation and the name of the folder where the create.log file is available.



Screenshot 7 — Successful end of Moses installation

1.5. Installing Windows-addins in MS Windows

If you want to use the Windows addins:

In MS Windows, go to the https://github.com/jladcr/Moses-for-Mere-Mortals/releases page and click the **Source code (zip)** button just under the **Windows-addins-1.043** release.

A dialog window will open with a **Save File** option. Select it and click the **OK** button. The **full download** will then start.

Uncompress the zipped file and install the *.exe files as for any other Windows program. Each of the installed programs has a readme.txt file that describes how to use it.

You are now ready to start the Demo!

2 — Running the Moses-for-Mere-Mortals Demo

After installing Moses for Mere Mortals following the instructions in Section 1, we recommend that you run the **Demo** — without changing any settings — to make sure that MMM is correctly installed and working properly and to have an idea of the steps involved and the relative time they take.

The small corpus needed for this **Demo** was automatically installed by the ***create*** script (no need for you to do anything) and is used by the other scripts, which are already configured accordingly.

We assume you are using the default location ***$HOME/Desktop/Machine-Translation*.** This is a "must" for you to run the **Demo** without changing any of the settings.

The **Demo** corpus is small (200,000 segments in the Portuguese and English languages) and the results of its processing cannot be seen as representative of the quality Moses can achieve especially if you consider that Moses can process corpora with several to many millions of segments.

Just for you to have an idea of what it takes to run this Demo:

The training of the **Demo** — not changing any of the default settings — took about 3 hours in PC1 and about 1h 30m in PC2.

The BLEU score — with all the defaults — was 71.88 (on a scale of 100; the score is presented in a scale of 0 to 1 as 0.7188).

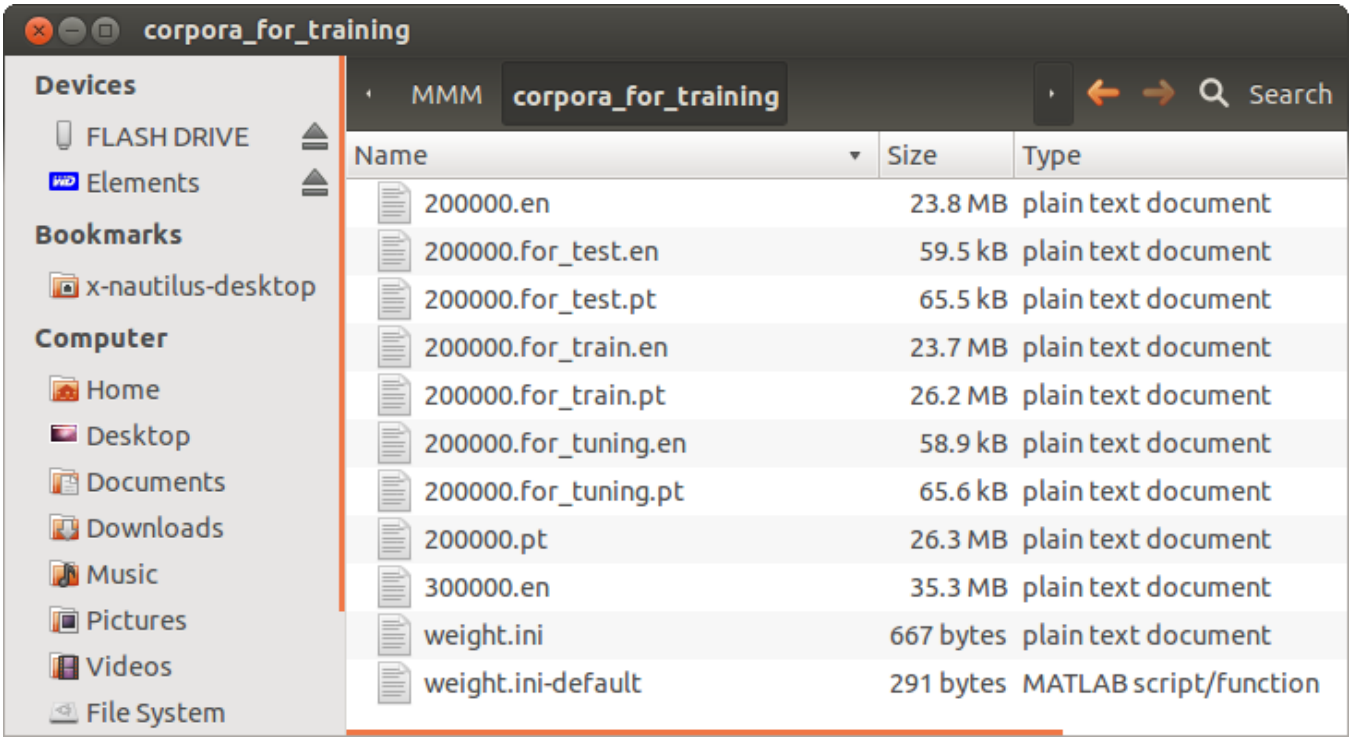
Moses will train this Portuguese-English corpus, which involves 200,000 segments for translation model training and 300,000 segments for language model building (target language: EN).

The test corpus already provided of 500 segments — extracted with the ***make-test-files*** script from the training corpus — is an in-domain test corpus. So don't get overoptimistic about the results of the translation with real documents which may be somewhat — or completely — out of domain.

The tuning corpus also provided has 500 segments and was also extracted with the ***make-test-files*** script.

2.1. First training with the Demo

All the corpora files mentioned above needed to run the **Demo** are already in the ***$HOME/Desktop/Machine-Translation/MMM/corpora_for_training*** folder.



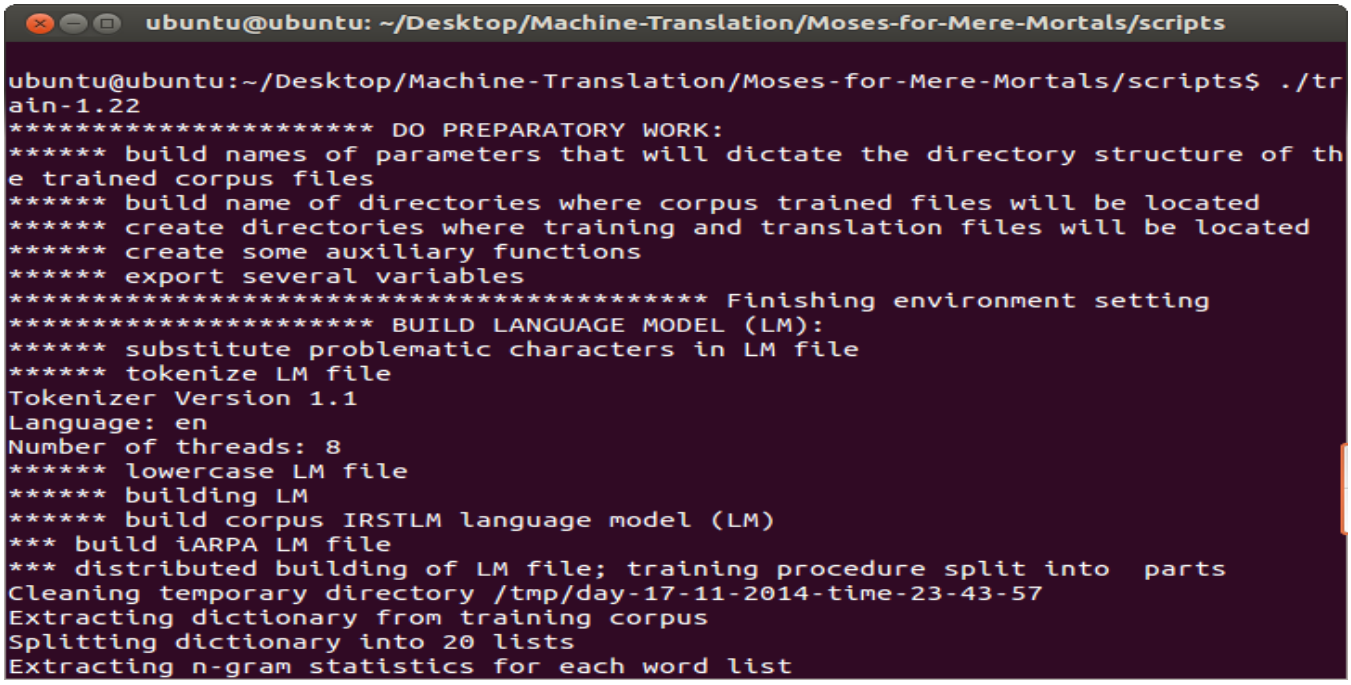
Screenshot 8 — Demo corpora (base corpus and corpora for train, test and tuning)

So you just have to:

1. Open the **Terminal**.
2. Position yourself in the folder where MMM was copied to by typing the path: ***$HOME/Desktop/Machine-Translation/Moses-for-Mere-Mortals/scripts***;
3. Launch the script with the command: ***./train****.

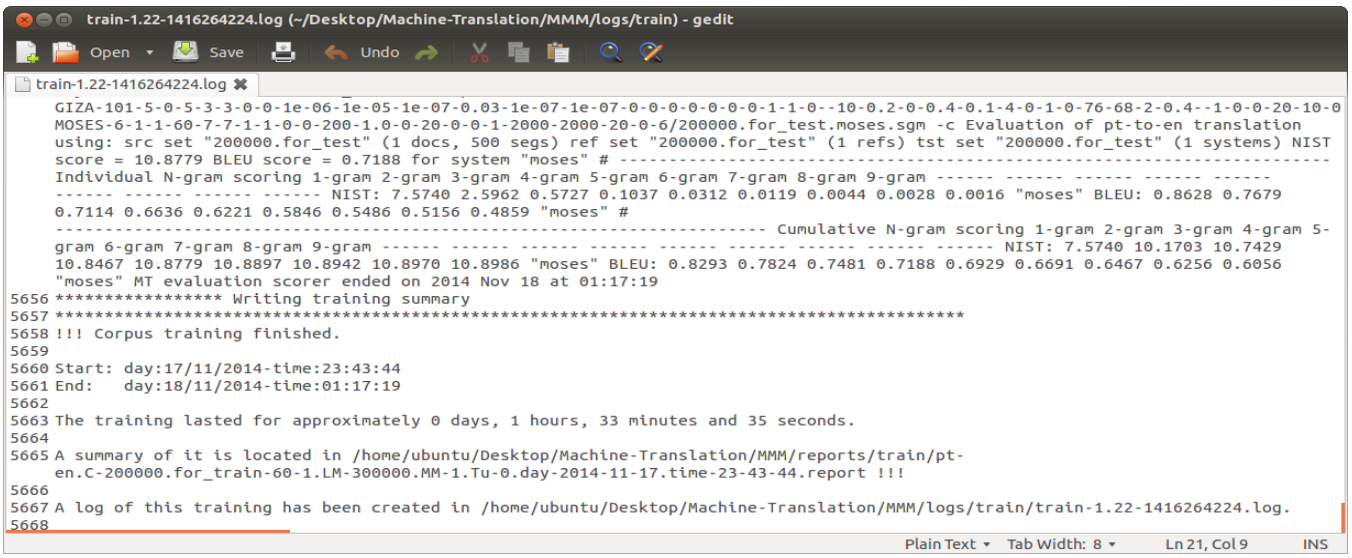
You can also double click on the name of the script if you want.

You will see the training progressing in the **Terminal** window.



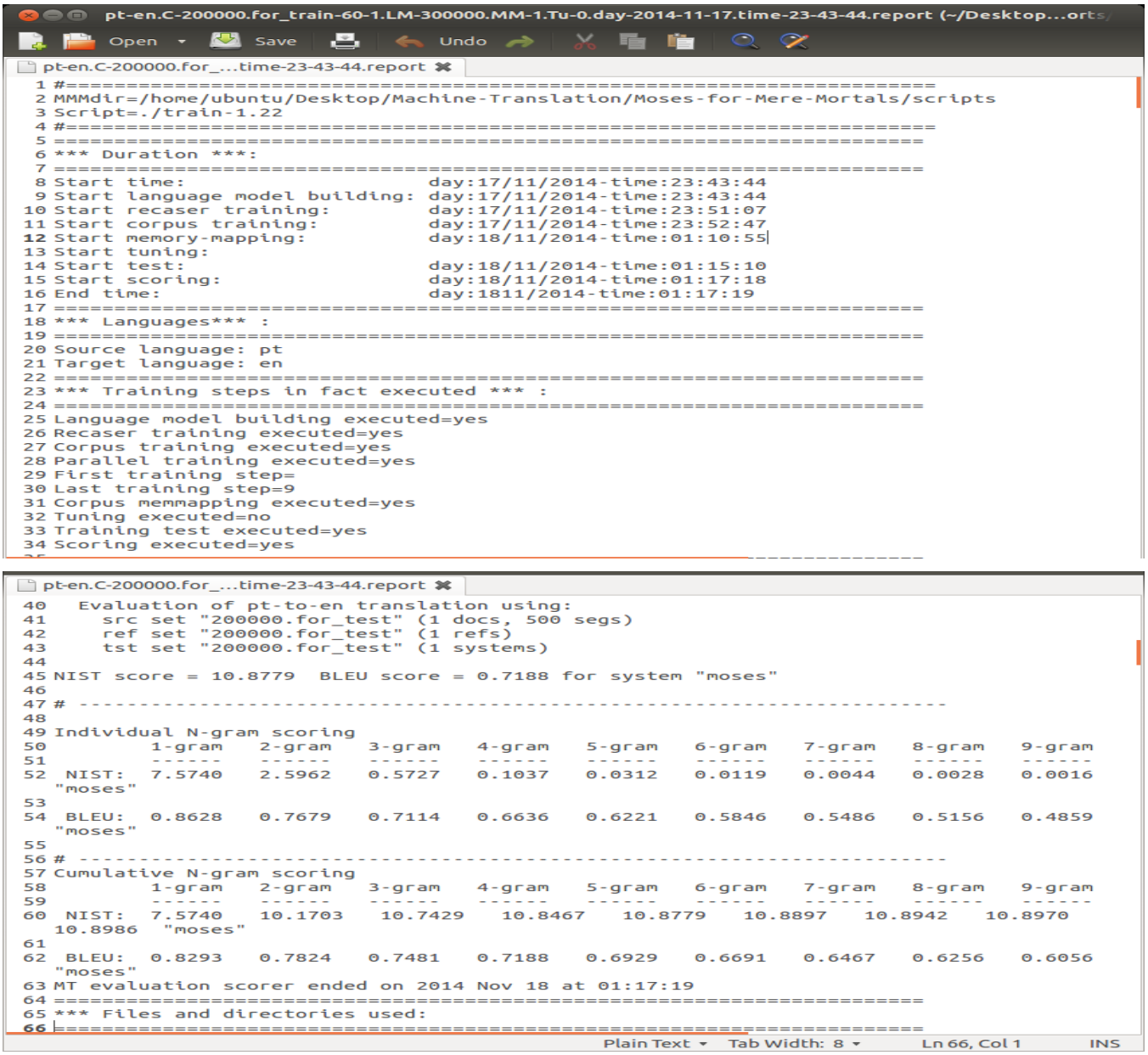
Screenshot 9 — Start of the training process

4. When the training is finished, in the **Terminal** you will — hopefully — see a message with information concerning that particular training. The same information is recorded in the training log.



Screenshot 10 — Log of the training of the Demo corpus

5. MMM also generates a report of the training with the main information. To see the report, in the **File Manager** go to the ***$HOME/Desktop/Machine-Translation/MMM/reports/train*** folder and open the only report that is there (as it is your first training).

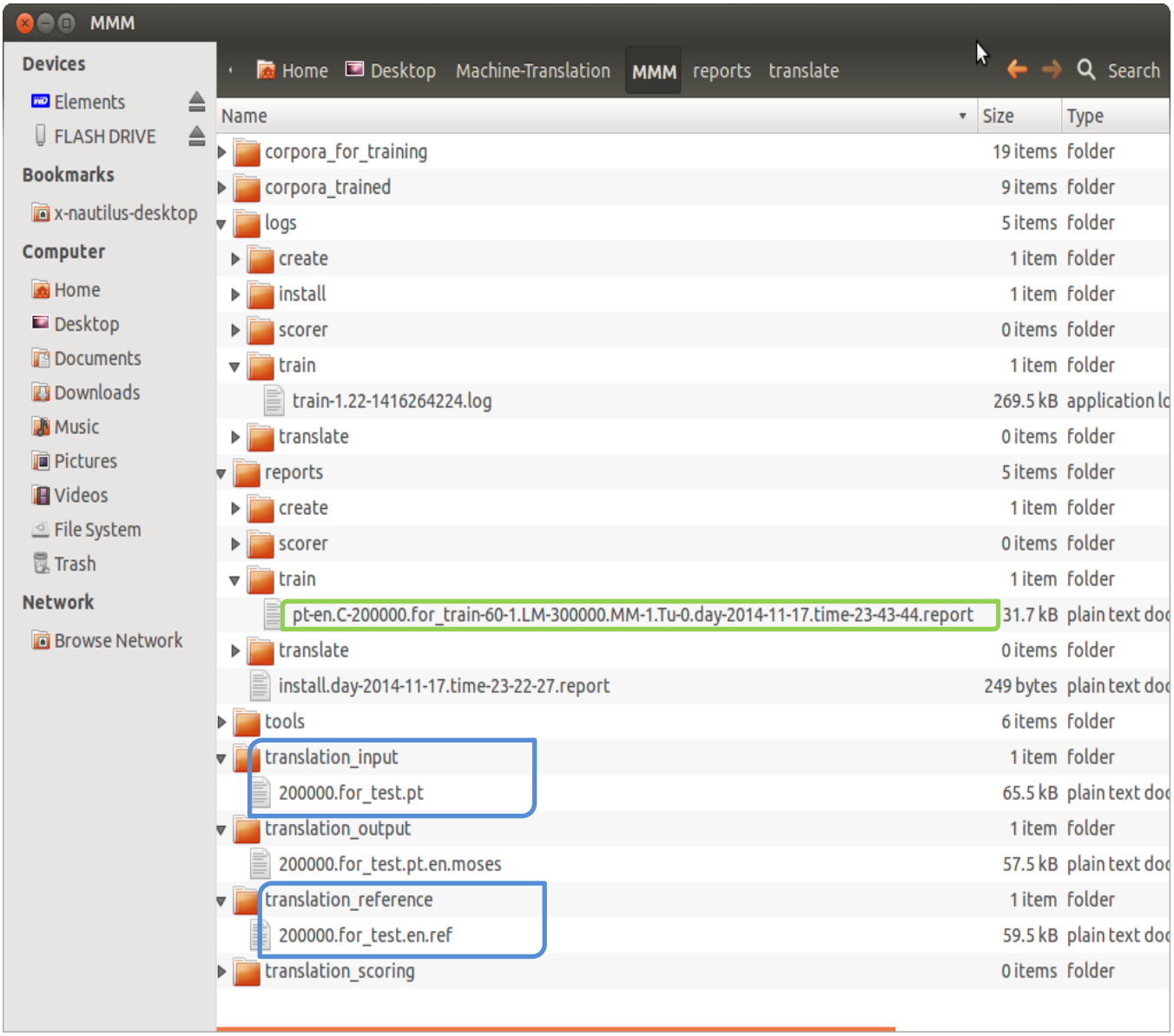


Screenshot 11 — Report of the training of the Demo with defaults

You are now ready to do your first translation with Moses!

2.2. First translation with the Demo

In the ***translation_input*** subfolder of the ***$HOME/Desktop/Machine-Translation/*MMM** folder, there is already 1 file to be translated as part of the **Demo** and a file with the reference human translation to be used for scoring MT output.



Screenshot 12 — MMM folder

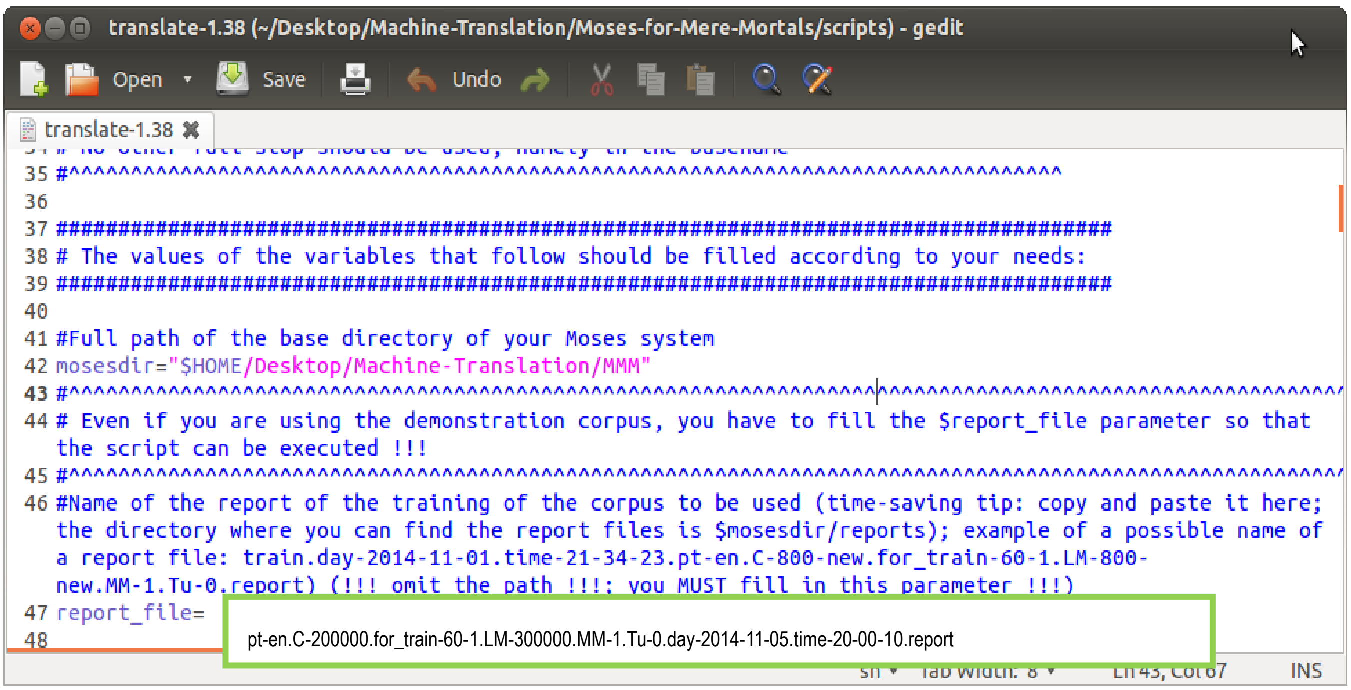
To be able to translate documents using this engine, you will first have to:

1) Copy the name of the report file in the ***$HOME/Desktop/Machine-Translation*/MMM/reports/train** folder, which is identified with the language combination, training corpus name, language model corpus name and time stamp.

The best way is to click to rename the report file and copy the whole name, including the extension).

This is a vital file and you are strongly urged not to change any of its contents nor its name otherwise the relevant training may become unusable.

2) Open the ***translate*** script and paste it in the ***report_file*** field.



Screenshot 13 — Translate script — *report_file* field to fill in with the name of the training

3) Run the **translate** script in the **Terminal** with the command: ./***translate-****.

4) After the translation is finished, see the Moses translation in the ***$HOME/Desktop/Machine-Translation/MMM/translation_output*** folder. The report of the translation is in the ***$HOME/Desktop/Machine-Translation/MMM/reports/translate*** folder

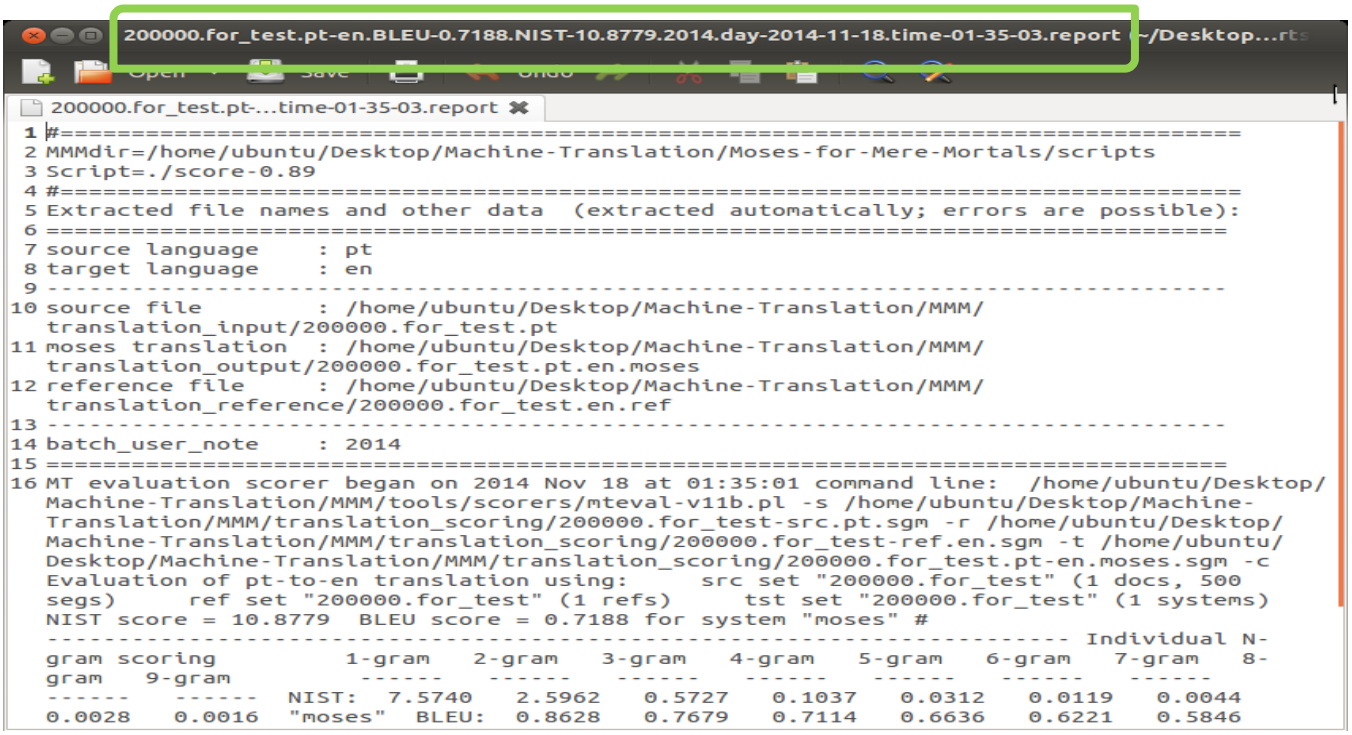
2.3. First scoring with the Demo

In the ***$HOME/Desktop/Machine-Translation/*MMM** folder, there is already 1 file in its ***translation_input*** subfolder and 1 reference file (human translation) in the ***translation_reference*** subfolder as part of the **Demo**. So, after running the ***translate*** script, just:

1) Launch the ***score*** script with the command: ***./score-*.***
2) When the scoring is finished, see the score file in the ***$HOME/Desktop/Machine-Translation/MMM/reports/scorer*** folder.

Notice the name of the score report (see screenshot 14). It already contains the most important information about the scoring: name of the test corpus, BLEU score, NIST score, **batch_user_note** and time stamp.

This way, if you do several scores of the same document for testing purposes, they will be ordered by BLEU score.



Screenshot 14 — Score report of the translation of the Demo testing corpus

If everything worked fine with the Demo, you can now start training your own corpora and translate your own documents.

So, now let's see in detail how to use Moses/Moses for Mere Mortals!

PART 2
– BASIC USE OF MOSES/MOSES FOR MERE MORTALS –

3 — Important preliminary information

The information in this Section is important for you to work with MMM.

As MMM may seem a bit confusing before you get "acquainted" with it, we recommend that you read this Section as it may save you a lot of time … and frustration!

But, if you want, go straight to Section 6 … at your own risk and peril if you are really an absolute beginner!

3.1. MMM default location and folders

MMM should be installed in the ***Machine-Translation*** folder created by you under ***$HOME/Desktop*** following the instructions in Part 1, at least for testing purposes, in order to train the **Demo** with the default settings.

In this Tutorial, we will refer the folder names omitting the full ***$HOME/Desktop/Machine-Translation*** path and just referring the name of the subfolders.

So, for example, by ***MMM/corpora_for_training*** we will mean the folder:

$HOME/Desktop/Machine-Translation/***MMM/corpora_for_training***

As you can also create a MMM installation anywhere you want in your computer (and therefore the path highlighted in blue can be different) and as you can also transfer trained corpora to another location or computer, it is simpler to refer the folders in this shortened way.

3.2. Scripts' names

So as to avoid having to update this Tutorial every time a script version is changed, the version numbers of the scripts have been omitted. For example, in most places we write ***train-**** or just ***train***, in order to refer to the ***train-1.22*** script, and the same for the other scripts.

3.3. Scripts

As you already saw if you ran the **Demo**, there are 7 script files: ***install***, ***create***, ***make-test-files***, ***train***, ***translate***, ***score*** and ***transfer-training-to-another-location.***

In these 7 scripts, you can define almost 100 parameters. However — be reassured — for the huge majority of them you don't need to do anything as they have default values and you can train your corpora and translate your documents and score MT output without changing them.

In the script files, there are 2 parts:

- The first part is where you must absolutely set the vital parameters for each operation and where you can also can define "interesting parameters" and other parameters if you want to change the defaults.
- The second part is marked:

```
################################################################################
# DON'T CHANGE THE LINES THAT FOLLOW ... unless you know what you are doing!
################################################################################
```

and you really should not change it — if you are a real mortal — otherwise MMM may not work properly.

The choices you make in the first part of the scripts will define the way your corpus will be trained or your documents translated or scored.

3.4. Vital parameters in the scripts files

There are a few parameters — that we call "**vital parameters**" — that you must define before running the relevant script. Most of the vital parameters relate to the specific corpora you want to train or use for translation.

Each parameter is preceded by a comment that describes its function, the MMM default value and, in some cases, the values that are allowable or the range of values recommended. They often consist of extracts of the Help files, readmes or manuals of the several packages used.

3.5. Interesting parameters in the scripts files

There are others parameters — that we call "**interesting parameters**" — for which you may want to try several options to test if you get better MT output for your language combination, your corpora and your documents for translation. But if you want, just forget about them as all have defaults!

For each parameter, there is a brief description and some comments based on our experience as "mere mortals". But take into consideration that the results may be somewhat — or very — different with different language combinations, different corpora and different types of documents to translate.

3.6. Number of cores used by Moses

In each script, there is a parameter called ***cores***. This is the parameter that defines the number of cores in your computer that you want Moses to use (Moses will be compiled to make better use of them). Leave this parameter empty if you want it to use all the cores available (the default).

Although usually you will be able to use your computer for "normal" applications — even during a training — you may want to leave 1 or more processors/cores for other applications.

In that case, in the field **cores**, define the number of cores you want MMM/Moses to use for each script/operation.

3.7. Changing the settings in the scripts files

To define the settings in the scripts files, you just have to:

1) Double click on the name of the file**.** The following options are displayed: "***Run in Terminal***", "***Display***", "***Cancel***" and "***Run***".
2) Select "***Display***" to edit the script file.
3) Change the settings you want.
4) Save the file.

If you want, you can change the name of the script, for instance to differentiate scripts to translate documents with different language pairs (Example: ***translate-EN-PT-1.38*** script and ***translate-FR-PT-1.38*** script).

3.8. Input and output files

Moses uses files in text format (UTF-8) both for input (either to train a corpus or translate a document) and output (to produce translations).

Therefore, all the files will have to be converted (if necessary) and you can use programs like Notepad++, Notepad2 or gedit to change that encoding.

3.9. Names of the files and of the languages

The names of the files and languages — which are used to create some directories names — should not include spaces or symbol characters, like asterisks, backslashes or question marks. Try to stick to letters, numbers, dot, and underscore if you want to avoid surprises.

Also avoid using a dash as the first character of a file name, because some Linux commands will treat it as a switch. If your files start with a dot, they will become hidden files.

3.10. Base corpus and base names

In MMM, we use the term **base corpus** to refer to a bilingual corpus (bitext) which is composed of 2 perfectly aligned files in UTF-8 format in the source and target languages.

With just these 2 files, you can automatically generate — with the ***make-test-files*** script — the corpora for training, testing and tuning.

We use the term **basename** to refer to the different corpora names that must be defined in the ***train*** script:

- **corpusbasename**: name of the corpus to trai the translation model (Ex: 200000.for_training.en and 200000.for_training.pt)
- **lmbasename**: name of the corpus to train the language model (in the target language) (Ex: 300000.en or 200000.for_training.en if you are using the base corpus to train this model)
- **recaserbasename**: name of the corpus to train the recaser model (in the target language) and usually the same used for training the language model (Ex: 300000.en or 200000.for_training.en if you are using the base corpus to train this model)
- **tuningbasename**: name of the corpus to be used for tuning (Ex: 200000.for_tuning.en and 200000.for_tuning.pt)
- **testbasename**: name of the corpus to be used to run the translation test at the end of a training (Ex: 200000.for_test.en and 200000.for_test.pt)

3.11. Instructions

In this Tutorial, we try to give not only operational instructions on how to run Moses via Moses for Mere Mortals but also to give you general information on how Moses works (in the **What Makes Moses Tick** Annex).

So, in each section, we assume that either you are already familiar with SMT/Moses concepts or that you have read that Annex to understand what some important parameter options are about... if you are interested in "playing" with them.

If not, don't worry! You can just accept the huge majority of the MMM defaults as defined in the script files.

3.12. Running MMM in the Terminal

 See also **QUICK-START-GUIDE.doc** in the docs subfolder.

Some users love to use the Terminal directly while others hate it. If you want, you can use the method described in point 3.13 below.

To run MMM script from the **Terminal**:

1) Open the **Terminal**.
2) If you do not have the icon in the left ribbon (the "**Launcher**"), select "***Dash home***", write "***Terminal***" and double click on it.
3) On the left ribbon, you can right click on the **Terminal** icon and select "**Lock to Launcher**" to easily access it in the future
4) Position yourself in the folder where MMM was copied to by typing the path for its location.

Example, assuming you have MMM in the default location:

cd ***/$HOME/Desktop/Machine-Translation/Moses-for-Mere-Mortals/scripts***

where $HOME is the name of your base folder. Normally, it is simply ***/home*** or ***home/{your login***}.

Launch the scripts with the command**: ./{*name of the script*}**

(Example: ***./create-1.43***).

3.13. Running MMM via the File Manager with the option Run in Terminal

If you want, you can run each script just by double clicking on its name and selecting ***Run in Terminal***.

It will immediately open the **Terminal** and run the script — and you can see it running in your screen. This is the easiest way to do it... for some users.

However, it has the disadvantage that the **Terminal** window will close when the operation is finished and you will not see (if you are not there looking at it all the time) if the script ran without problems or not.

This is really not a problem as with any of the scripts there is always a log and a report — in the ***MMM/logs*** and the ***MMM/reports*** folders — where you can check if everything was alright.

To run the scripts from the **File Manager**:

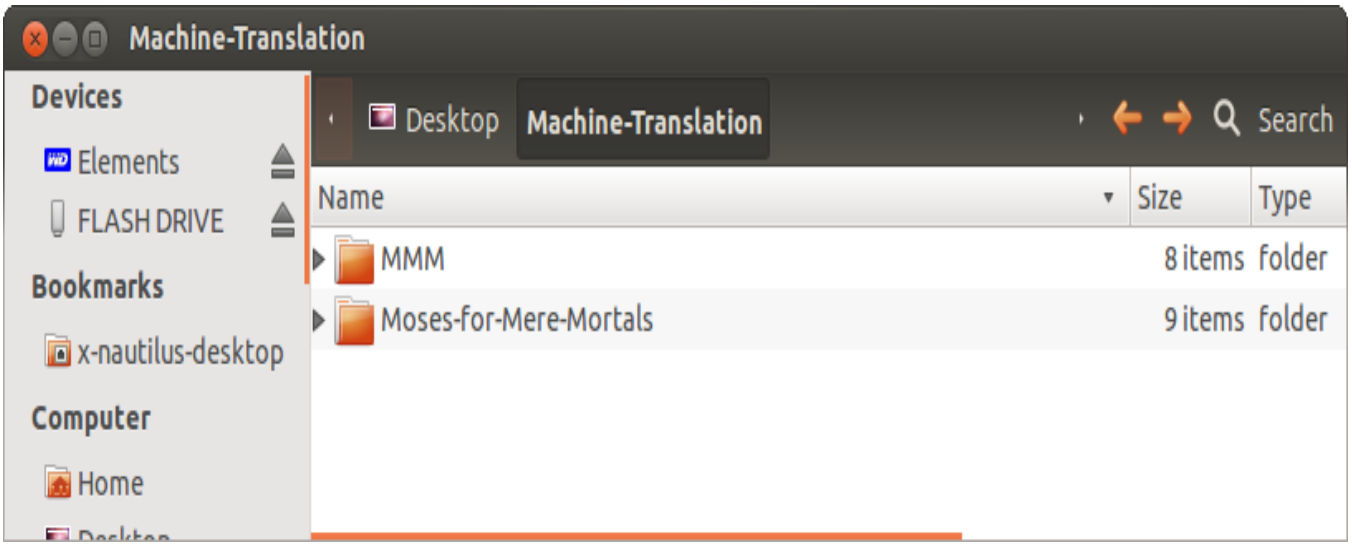
1) Double click on the name of the script and you should see 4 options.
2) Click on ***Run in Terminal*** and the **Terminal** window will open. The window will close automatically when the operation ends.

 In this Tutorial, to indicate the running of a script we just say **Run in Terminal** and you do it the way you prefer.

4 — How MMM is organised

In this Section is presented the general structure of Moses for Mere Mortals which will be explained in detail in the following Sections.

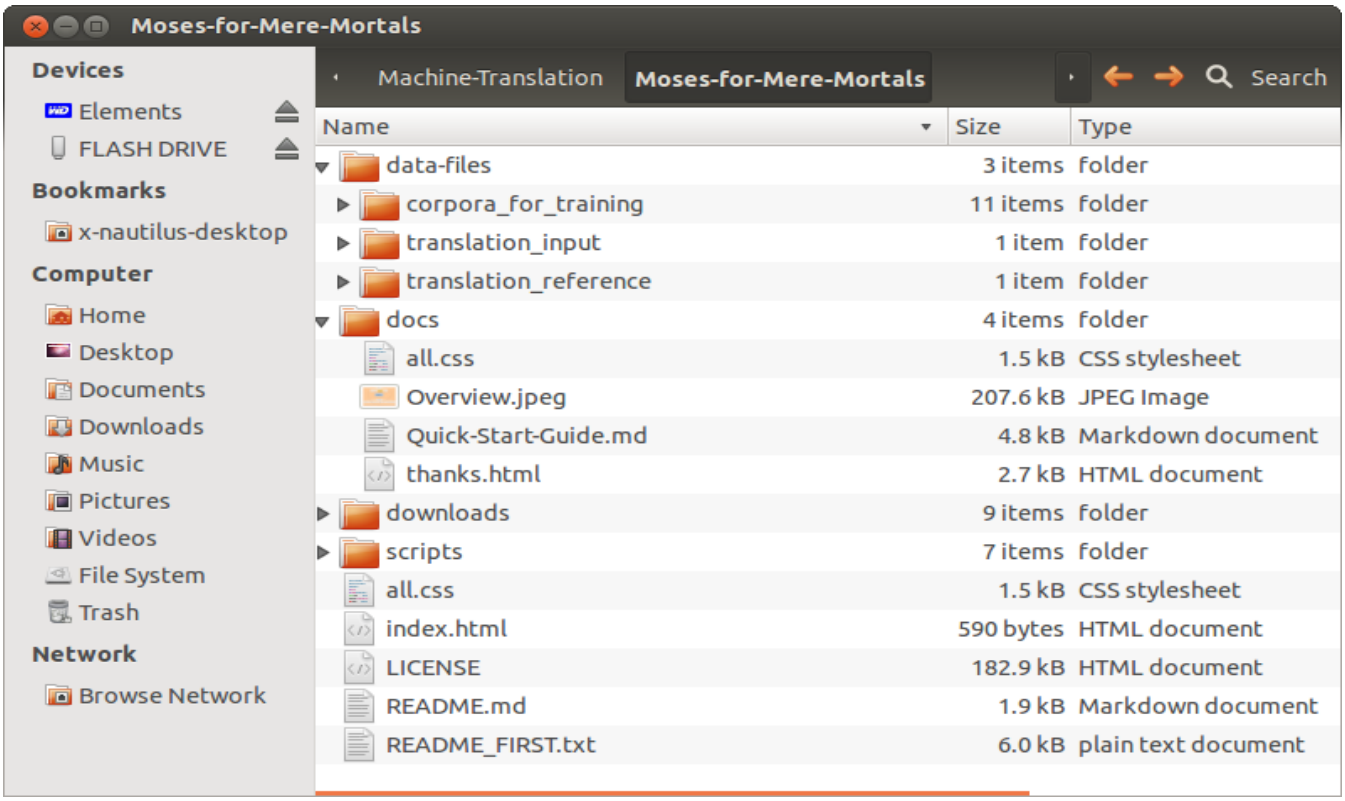
When MMM is installed, the following folder structure is created:



Screenshot 15 — The *Machine-Translation* folder

4.1. *Moses-for-Mere-Mortals* folder

This folder contains several subfolders, of which the ***scripts*** folder is the really important one for you.

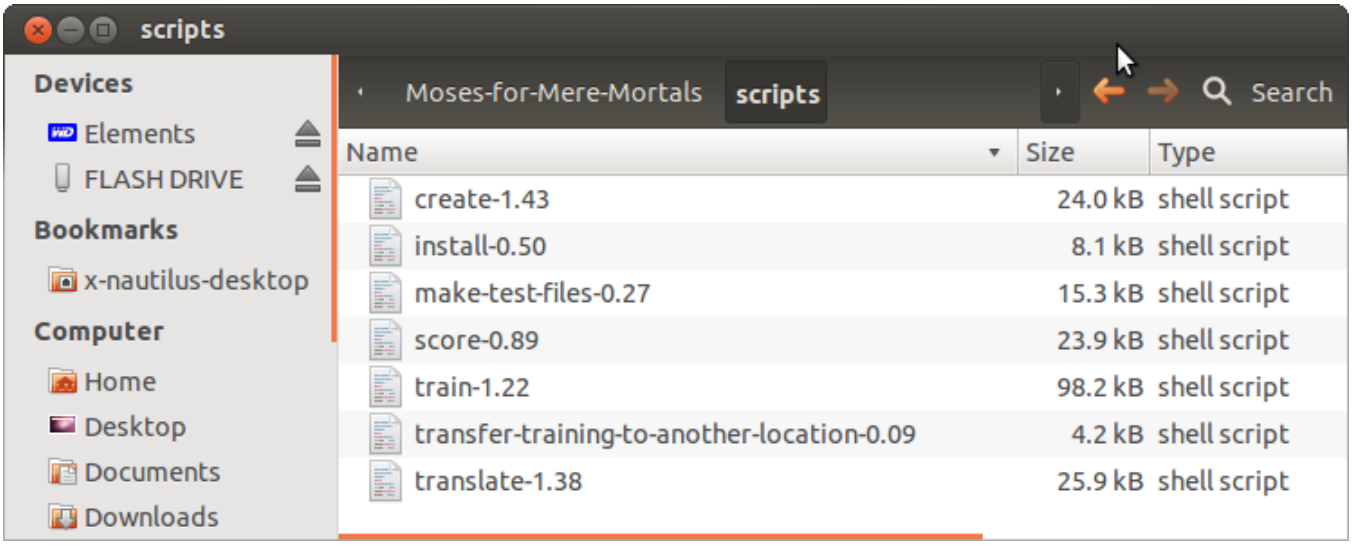


Screenshot 16 — The *Moses-for-Mere-Mortals* folder

1) The ***data-files*** folder with the **Demo** files — **200000.for_train, 200000.for_test** and **200000.for_tuning** — so that you can test the ***train*** script, and the ***translation_input*** and ***translation_reference*** files for you to test the ***translate*** and ***score*** scripts.
2) The ***docs*** folder with the Overview with the structure of MMM, the Quick-Start-Guide and the Thanks file … and to where you can copy this Tutorial if you want.
3) The ***downloads*** folder with all the packages that are needed for MMM to run.
4) The ***scripts*** folder — which is the one you are going to use all the time — to run MMM.
5) The ***README_FIRST*** file with a brief explanation about how MMM works.

4.1.1 The *scripts* subfolder

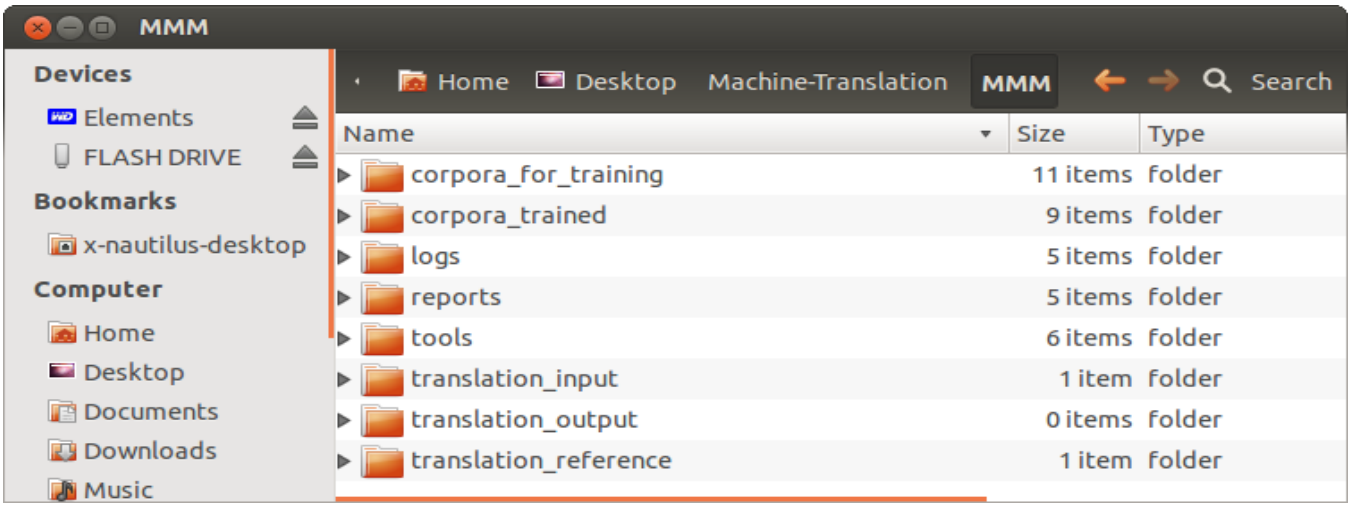
This is the folder — already mentioned — in which you find the ***scripts*** (executable files) that you will use all the time to work with Moses.



Screenshot 17 — The *scripts* folder

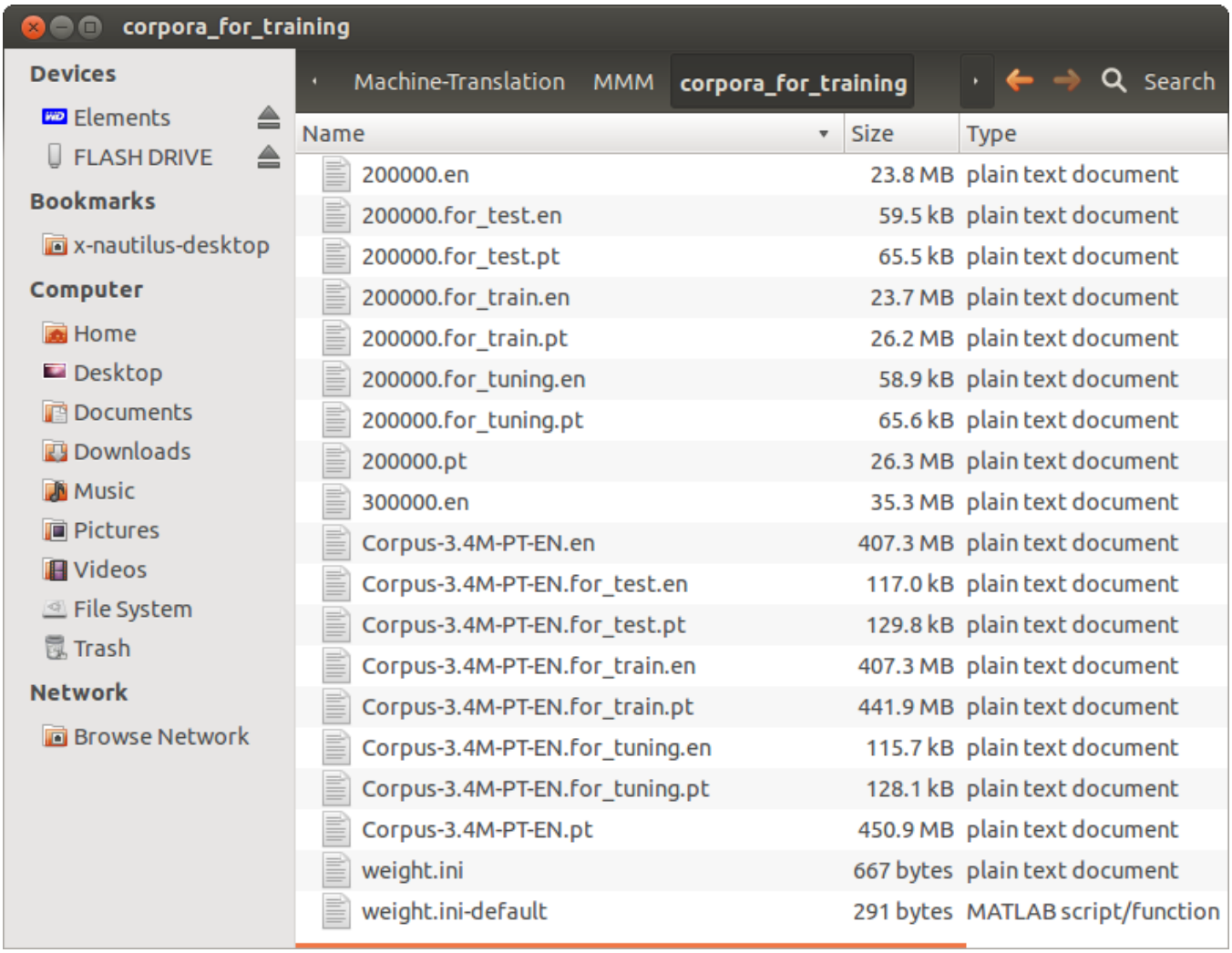
4.2. *MMM* folder

This is the folder where Moses and all the packages needed are installed and where you will manage your training, test and tuning corpora, copy your documents to translate, get your MT translations and automatically score them.



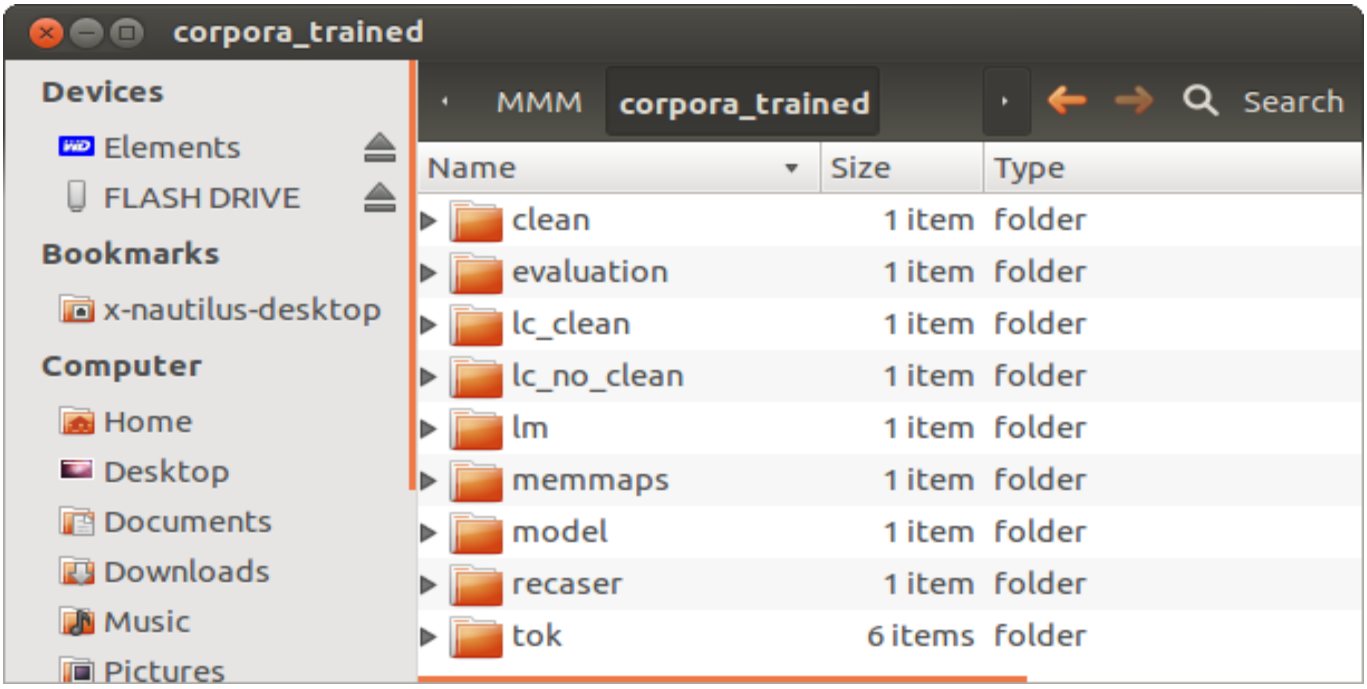
Screenshot 18 — The MMM folder

1) The ***corpora_for_training*** folder, where you place the corpora you want to train.



Screenshot 19 — The corpora_for_training folder, in this example with the Demo corpus and the Corpus-3.4M-PT-EN

2) The ***corpora_trained*** folder, where MMM stores all the training files of the corpora you train in a MMM installation (it will be created just after you start your first training).

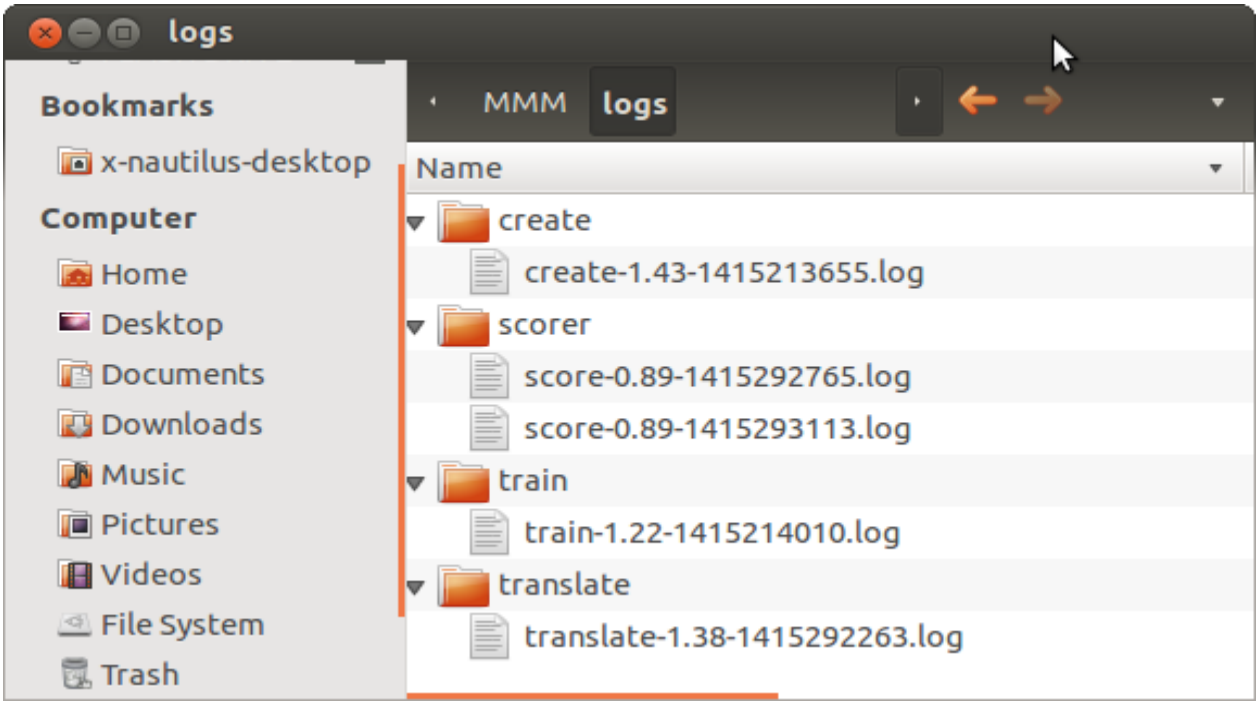


Screenshot 20 — The corpora_trained folder

You can train several corpora — with different language combinations or different settings— in the same MMM installation.

This is a vital directory and you are strongly urged not to change any of its contents. If you do, you risk destroying the training(s).

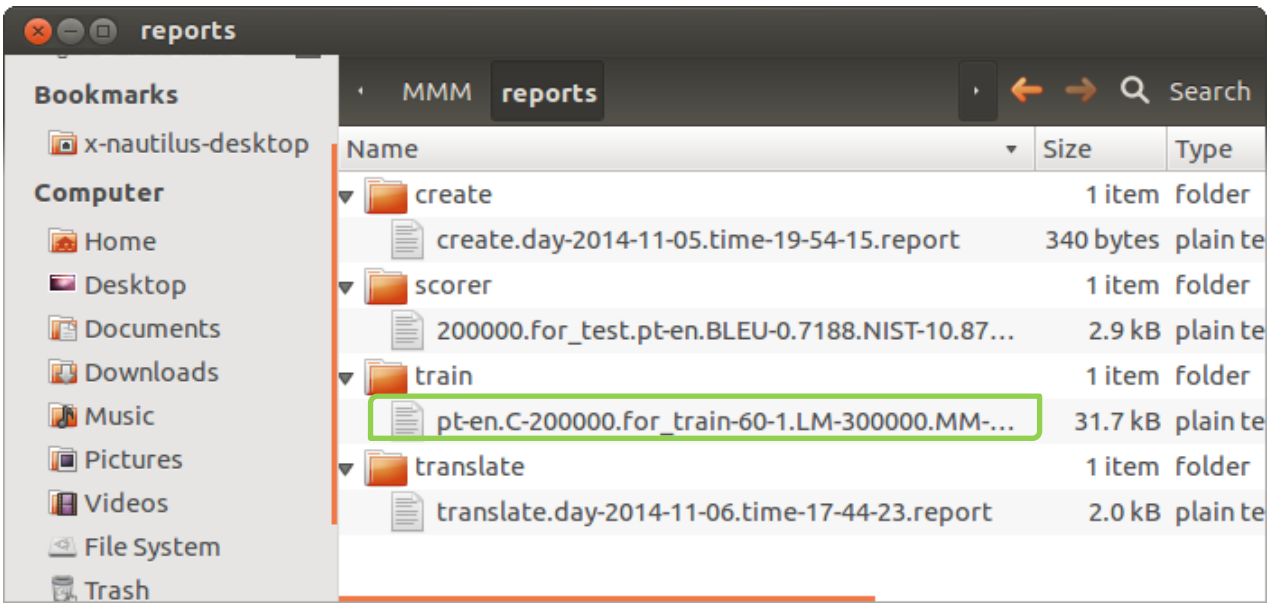
3) The ***logs*** folder, which contains the logs of the creation of a MMM installation, of the training(s) you have done and of the (set of) documents you have translated or scored.



Screenshot 21 — The *logs* folder

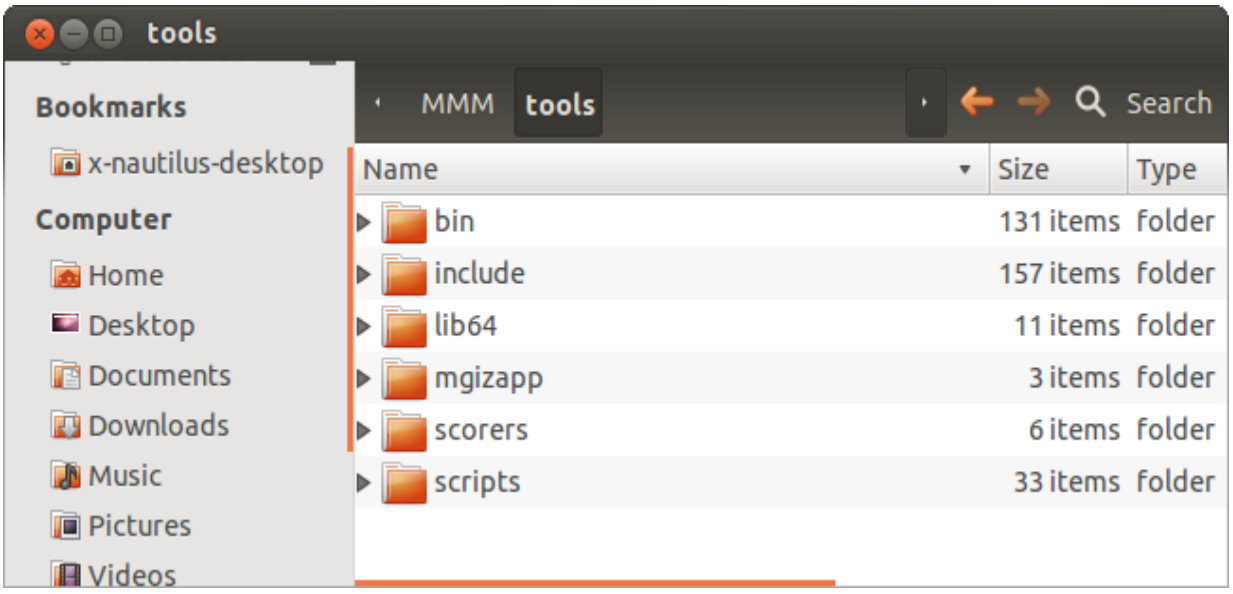
4) The ***reports*** folder, which contains the reports of the creation of a MMM installation, of the training(s) you have done and of the (set of) documents you have translated or scored.

Of special importance are the training reports. It is the name of the corresponding files that you must copy to the ***translate*** script in order to produce MT translations with that particular training corpus (called 'engine').



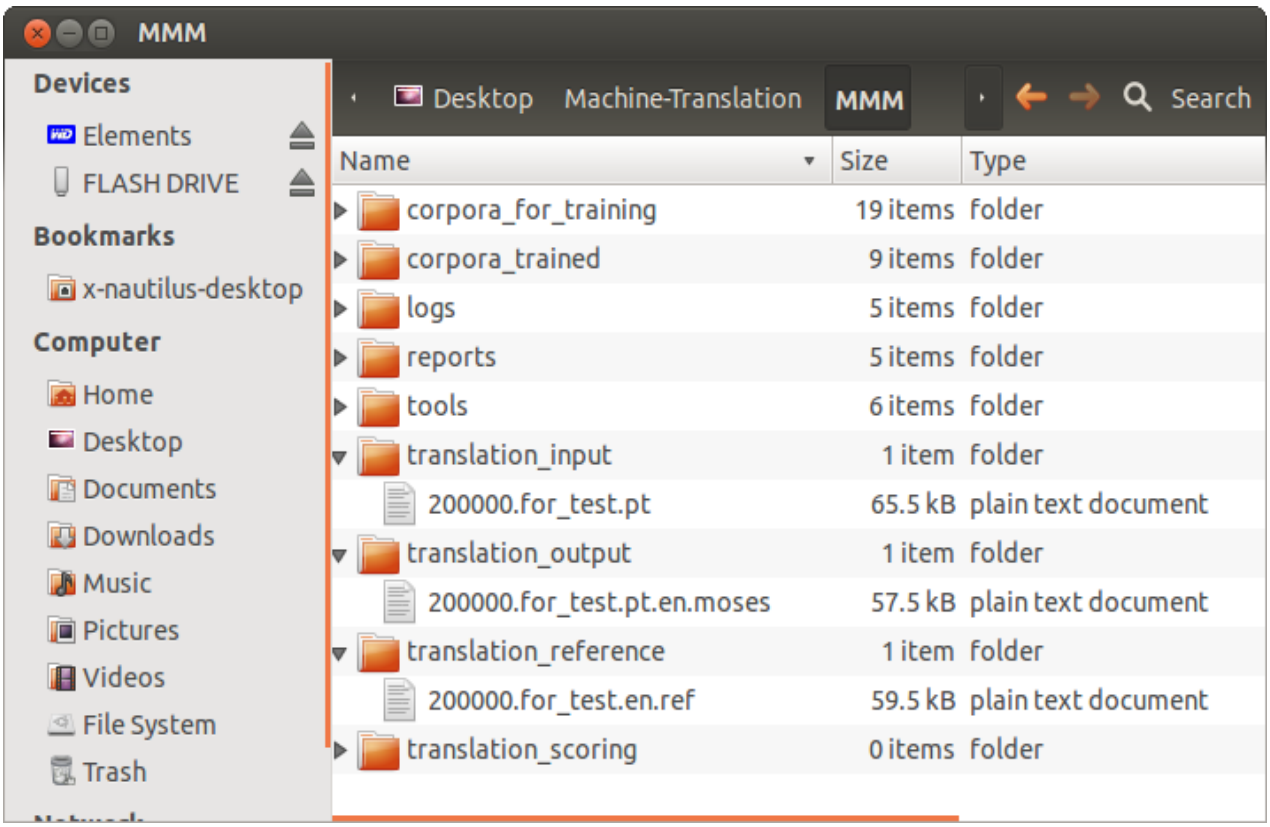
Screenshot 22 — The *reports* folder

5) The ***tools*** folder, which contains the packages Moses needs to run.



Screenshot 23 — The *tools* folder

This is a vital directory and you are strongly urged not to change any of its contents.



Screenshot 24 — The translation input, output and reference folders

6) The ***translation_input*** folder, where you will place the documents you want to translate or score.
7) The ***translation_output*** folder, where Moses translations are generated.
8) The ***translation_reference*** folder, where you should place your reference documents if you have human translations that you want to use to score MT output. The reference file must have the same **basename** as the file in the ***translation_input folder***, followed by the language extension and the **.ref** extension.
9) The ***translation_scoring*** folder is a folder for temporary files.

5 — Corpora needed

If you want to train your own corpora, you must first have them in a format accepted by Moses.

5.1. Need for strictly aligned corpora files

If you do not have separate and strictly aligned files (one in the source language and another in the target language), but you do have *.TMX translation memory files — either your own or from corpora available on the Internet — you can use the ***EXTRACT_TMX_CORPUS_1.043.EXE*** Windows addin to convert such *.TMX files into corpora files that Moses can use.

Be ***very sure*** that the corpus files you use are **strictly aligned**, otherwise you risk getting quite puzzling errors. At the very least, check the number of lines of the source language file and of the target language files, which should be exactly the same.

In order to do this, either:

1) In the **File Manager**, double click on the name of each file and open it in **gedit** or any other application to check that they both have the same number of lines and that the last line of the source language file corresponds to the last line of the target language file,

   or

2) Open the **Terminal**, change the directory to where you have the corpora folder and type in the **Terminal** the commands:

   - **wc –l {*name_of_source_language_file*}** (to check the source file)

     and

   - **wc -l {*name_of_target_language_file*}** (to check the target file)

The training, testing and tuning files — if generated from your base corpus using the ***make-test-files*** script — are strictly aligned and you don't need to check them.

5.2. Base corpus files needed to train an MT engine

To train an MT engine, you only need to have a bilingual corpus, which we call a **base corpus** (in our example, Portuguese-English) and to:

a) Copy — to the ***MMM/corpora_for_training*** folder — those 2 perfectly aligned UTF-8 files with the same name and with the extension for the source and target languages.

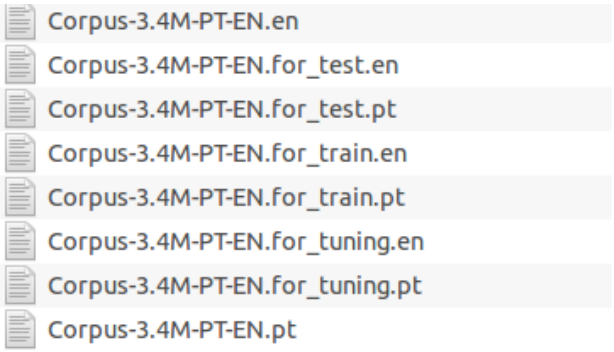
Example: Corpus-3.4M-PT-EN.pt and Corpus-3.4M-PT-EN.en.

b) Use the ***make-test-files script*** if you want MMM to generate automatically — from your base corpus — 2 files for training, 2 files for testing and 2 files for tuning (see Section 6).

This is not mandatory, but it will allow you to quickly have a general idea of MT performance level.

MMM will use the bilingual corpus to generate the files for training, testing and tuning — without any need for you to provide other corpora — and to train all the necessary models (namely, Language Model, Translation Model and Recaser Model).

After running the ***make-test-files*** script, in the ***MMM/corpora_for_training*** folder you will then have the following files, in the case of the corpus used as example:



5.3. Using other corpora for the LM, testing and tuning

If you only have a corpus in the source and target language, you can skip the next subsections.

But if you have — or want to process — different corpora for the Language and Recaser Models, for testing and/or for tuning, read the next subsections.

5.3.1. File for the Language and Recaser Models

This is a corpus only in the target language and can be:

- The target language file used for the training or
- A (much) bigger corpus — as monolingual data is easier to get — which can be composed of the target language file you already have and to which you can add more data.

So, if you have a larger corpus you want to use instead of the target language file used for training, just copy it to the ***MMM/corpora_for_training*** folder

This corpus doesn't need to have any other extension except the language abbreviation.

Example: Corpus-3M-PT-EN.en.

5.3.2. Files for testing and tuning

You can also use a testing corpus and a tuning corpus that you already have and that is not contained in the training corpus.

For that you need:

1) 2 files perfectly aligned of the size you want to use as testing corpus
2) 2 files also perfectly aligned of the size you want to use as tuning corpus.

Take into consideration that tuning is a step that may take a long time, depending on the options you choose. A 500 to 2000 segment corpus is generally considered enough.

To use them in a particular training, you must rename these files with:

a) The extension ***.for_test*** followed by the language abbreviation
b) The extension ***.for_tuning*** followed by the language abbreviation.

In this case, you must check that the testing and tuning files you have are strictly aligned.

5.3.3. Set of files needed for training

Summing it up, for an example corpus named Corpus-2M-PT-EN with a corpus for the language and recaser models in the target language named Corpus 5M-PT-EN, a corpus for testing named Test1000 and a corpus for tuning named Tuning1000 to be used to train an engine PT-EN, you must have the following 7 files:

- To train the Translation Model: Corpus-2M-PT-EN**.for_training**.en and Corpus-2M-PT-EN.**for_training**.pt
- To train the Language and Recaser Models: Corpus-5M-PT-EN.en
- To be used for testing: Test1000-PT-EN.**for_test**.en and Test1000-PT-EN.**for_test**.pt
- To be used for tuning: Tuning1000-PT-EN.**for_tuning**.en and Tuning1000-PT-EN.**for_tuning**.pt

These files must be in the ***MMM/corpora-for-training*** folder before you launch the ***train*** script.

5.4 — EU and other corpora freely available on the Internet

If you do not have tmx files with previous translations, or if you want to complement your own memories to increase their coverage, there are corpora freely available on the Internet that you can use.

When you have your own corpora and any of the others below converted in the format to be used by Moses (2 perfectly aligned UTF-8 files) you can merge them using the Linux merge command.

For example, to merge all the files with the **.en** extension, just place them in a folder and in the **Terminal** run the command: **cat *.en > all.en**. You will get a file which contains all the segments of the files in that folder, rigorously keeping its order. You can do the same to the other language files and get 2 merged files perfectly aligned!

6 — Generating corpora for training, testing and tuning from a base corpus

Script: ***make-test-files*** in the ***Moses-for-Mere-Mortals/scripts*** folder

If you have your own test and tuning corpora, (which are not contained in the corpus for training) skip this Section and just copy them to the ***MMM/corpora_for_training*** folder.

In this Section we explain how to automatically extract testing and tuning corpora from your base corpus.

The ***make-test-files*** script generates the test corpus and the tuning corpus, by dividing the original corpus into X sectors (a parameter that you can define) and then randomly selecting Y segments (another parameter you can define) in each sector. All the selected segments will have different line numbers (no line can be chosen more than once).

It also generates a new corpus for training from which these lines are erased. For testing purposes, this procedure offers a better guarantee that the segments used for testing the trained corpus are more representative of all the styles and contexts of the corpus being used than they would be if you would arbitrarily choose the same number of consecutive segments somewhere in the input files.

However, as many segments may occur several/many times in the original corpus, identical segments can still occur in the new corpus training files.

If a segment occurs 1000 times in the original corpus and one line with it is erased from the training corpus to be used in the test corpus, there will still be 999 lines with an identical segment in the training corpus.

If you want to compare the relative results of a change in training parameters, you should run the training test before and after the change in parameters with the same set of test files. Run ***make-test-files*** scripts only once and use the test files it creates to test both (or several) trainings.

The script will create the source and target files with the same base name and the extensions: ***.for_test**; ***.for_tuning**; and ***.for_train** (6 files in total), which are automatically placed in the **corpora_for_training** folder.

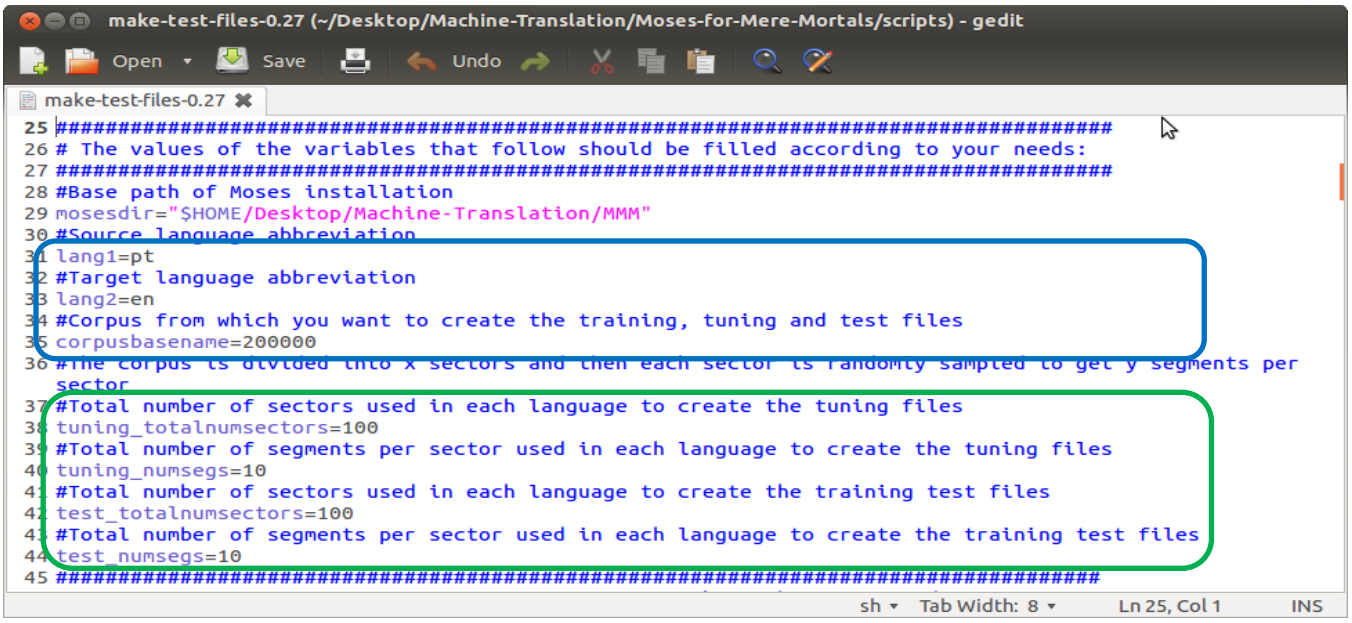
If you don't change the defaults, the script will generate a corpus for test with 1000 segments, a corpus for tuning with 1000 segments and also a corpus for training from which those segments have been erased.

For the original corpus called Corpus-3.4-PT-EN we have been using as an example, the following corpora will be generated and these are the names which must be afterwards defined in the ***train*** script:

- Corpus-3.4-PT-EN.for_train.pt
- Corpus-3.4-PT-EN.for_train.en
- Corpus-3.4-PT-EN.for_test.pt
- Corpus-3.4-PT-EN.for_test.en
- Corpus-3.4-PT-EN.for_tuning.pt
- Corpus-3.4-PT-EN.for_tuning.en

6.1. Vital parameters

1) ***lang1*** **—** source language
2) ***lang2*** **—** target language
3) ***corpusbasename*** **—** the name of the corpus from which you want to extract the test and tuning corpora and which you must have copied to the ***MMM/corpora_for_training*** before running the ***make-test-files*** script.



Screenshot 25 — *Make-test-files* script

6.2. Other parameters

If you use very small corpora to test MMM, take into consideration that the totalnumsectors x numsegs must be less than the total number of segments that exist in corpus to be trained.

6.2.1. tuning_totalnumsectors

Number of sectors in which the original corpus will be divided in order to create separate files (source and target) for the tuning corpus. They will be generated with the suffix .for_tuning and the language abbreviation. The default is 100.

test_totalnumsectors=100

6.2.2. tuning_numsegs

Number of segments pseudorandomly searched in each sector in the file used for running the training test in order to create a separate file for tuning.

tuning_numsegs=10

6.2.2. test_totalnumsectors

Number of sectors in which the original corpus will be divided in order to create separate files (source and target) for the training test. They will be generated with the suffix .for_test and the language abbreviation. The default is 100.

test_totalnumsectors=100

6.2.3. test_numsegs

Number of segments pseudorandomly searched in each sector in the file used for running the training test in order to create a separate file for testing. The default is 10.

test_numsegs=10

7 — Training basically with the default configuration

Script file: ***train*** in the ***Moses-for-Mere-Mortals/scripts*** folder

The ***train*** script is where you can define the settings for a particular training.

All the settings (including the vital parameters) are filled in so that you can train the PT-EN **Demo** corpus (200,000 segments) provided.

But to train your own corpora you absolutely have to fill in the **vital parameters** indicated below.

In the ***train*** script you can define a substantial number of parameters — 70 in fact — if you want to.

See Part 3 for information on exploring training options — interesting parameters — if you want to test training variations that may — or may not — produce better MT output depending on your language combination and corpora.

If you want to change any of the other parameters, see also the notes in the ***train*** script for each parameter and/or consult the Moses Manual as they are not explained in this Tutorial.

In the present Section we assume that you have read the previous section 5 on Corpora, that you have generated the train, test and tuning corpora with the ***make-test-***files script or that you have your own train, test and tuning corpora.

7.1. Vital parameters

The vital parameters are:

1) ***lang1*** and ***lang2*** in Section 1 — Languages — of the script
2) ***corpusbasename, lmbasename, recaserbasename, testbasename*** and ***tuningbasename*** (if you want to use tuning) in Section 2 — Files of the script.
3) ***tuning*** in Section 6 — Tuning Parameters of the script (optional).

7.1.1. lang1 and lang2

You can train any language pair you want using Moses via the MMM scripts.

It is here that you define the source language — ***lang1*** — and the target language — ***lang2*** — with a 2 character extension.

Example: lang1=pt and lang2=en

7.1.2. corpusbasename, lmbasename, recaserbasename, testbasename and tuningbasename files

As explained in Section 5, you must have the corpora copied to the ***MMM/corpora_for_training*** folder**.**

In the ***train*** script, you must fill in the **basenames** (without language abbreviation) as follows:

```
train-1.22
#^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
#Basename of the corpus placed in $mosesdir/corpora_for_training (the default value refers to the 2 files 200000.for_train.en and 200000.for_train.pt, whose basename is 200000.for_train)
corpusbasename=Corpus-3.4M-PT-EN.for_train

#Basename of the file used to build the language model (LM), placed in $mosesdir/corpora_for_training (!!! this is a file in the target language, having $lang2 as its extension; the extension should be omited when filling this parameter !!!)
lmbasename=Corpus-3.4M-PT-EN

#Basename of the recaser training file, placed in $mosesdir/corpora_for_training (!!! this is a file in the target language, having $lang2 as its extension; the extension should be omited when filling this parameter !!!)
recaserbasename=Corpus-3.4M-PT-EN

#Basename of the tuning corpus, placed in $mosesdir/corpora_for_training (the default value refers to the 2 files 200000.for_tuning.en and 200000.for_tuning.pt, whose basename is 200000.for_tuning)
#For a real corpus, other than the demo corpus, try to use files with 1000-2000 segments
tuningbasename=Corpus-3.4M-PT-EN.for_tuning

#Basename of the test set files (used for testing the trained corpus), placed in $mosesdir/corpora_for_training (the default value refers to the 2 files 200000.for_test.en and 200000.for_test.pt, whose basename is 200000.for_test)
#For a real corpus, other than the demo corpus, try to use files with 1000 segments
testbasename=Corpus-3.4M-PT-EN.for_test
sh ▾   Tab Width: 8 ▾   Ln 50, Col 1   INS
```

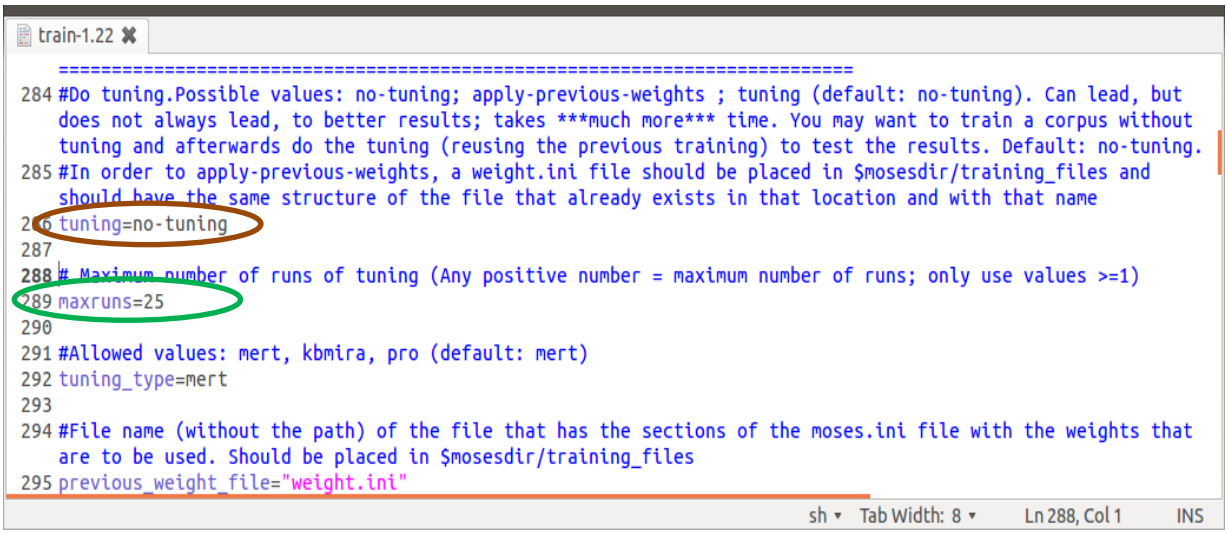
Screenshot 26 — Defining the corpora *basenames* in the *train* script

The basenames of the language model and the recaser models can be the same as the corpusbasename, if you are not using different (larger) corpora to train these models.

7.1.3. tuning

Typically, the results of a tuned engine should be better, but we suggest that you first train a corpus without tuning (the default) and afterwards retrain it with tuning to see the difference.

As MMM reuses a full previous training with the same parameters, if you launch a new training with only the tuning parameter changed, it will only perform this operation.



Screenshot 27 — Selecting the tuning option in the *train* script

If you want to do the tuning right away, just change the parameter (by default=no-tuning) to tuning:

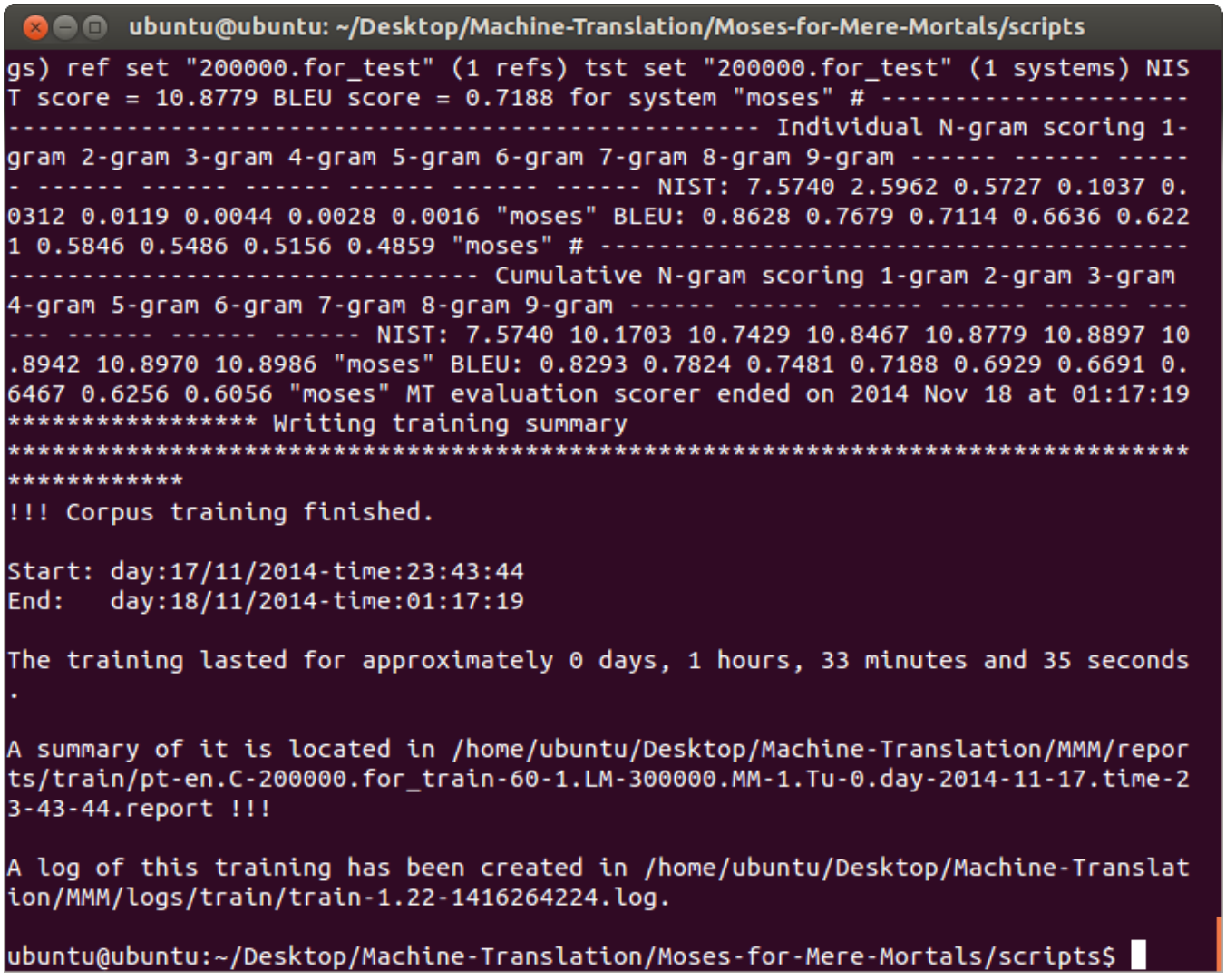
tuning=tuning

For more information on tuning see Section 12.1.2.

7.2. Run the training

After defining/checking these vital parameters, you can start the training just as you did for the **Demo** running it in the **Terminal** or by double-clicking on the ***train**** script in the **Moses-for-Mere-Mortals/scripts** folder and choosing **Run in Terminal**.

When the training ends, in the **Terminal** you will see this information:



Screenshot 28 — End of the Demo training in the Terminal

7.3. Training report and log

When the training is finished, it is stored in the ***MMM/corpora_trained*** subfolder.

The training report is stored in the ***MMM/reports/train*** subfolder There you have the relevant information concerning each training — the parameters used to train the corpora –, the time it took and the automatic scores of the test corpus (if you had one).

Do not change, rename or delete this file as it contains all the information necessary to use that training afterwards for translation purposes.

A log of the training (all the steps you see in the **Terminal**) is stored in the ***MMM/logs/train*** folder.

You will only need it if something goes wrong with a particular training and you want to check error messages or send it to someone for help.

And that's all!

8 — Translating your documents using the defaults

Script file: ***translate-**** in the ***Moses-for-Mere-Mortals/scripts*** folder

As explained in the previous section, when the training is finished you are ready to translate your own documents.

You will have first to convert them to a format that can be used by Moses (text with UTF-8 encoding).

You can translate 1 document or a batch of as many documents as you want.

In the ***translate*** script, you can change some parameters as to the way your documents will be translated. But for the moment just accept the defaults.

8.1. Vital parameter

To translate your documents you must first define — in the ***translate*** script — the name of the training you want to use as you already did for the **Demo** (see Section 2.2):

1) In the ***/MMM/reports*** folder, copy the name of the report file (Example: pt-en.C-200000.for_train-60-1.LM-300000.MM-1.day-27-10-14-time-00-30-45.txt).

   The easiest and surest way is to click on the name of the file, select ***rename*** and (being very careful not to delete the file or change its name) copy its name (including extension).

2) In the ***Moses-for-Mere-Mortals/scripts*** folder, open the ***translate*** script by clicking on ***Display***.

3) Copy the name of the report to the ***report_file*** field.

If you want, you can change the name of the script, for instance to differentiate scripts to translate documents with different language pairs (Example: translate-EN-PT-1.38 script and translate-FR-PT-1.38 script).

If you anticipate that you will want to simultaneously translate documents with different language pairs — and your computer is up to it — you can have two separate installations of MMM with the training of each language pair and you can run the ***translate*** script of each of the different installations at the same time.

8.2. Prepare the original documents

You also have to prepare the original documents by converting them to a format accepted by Moses. To do that:

1) Convert the original documents you want to translate to the UTF-8 format, if necessary using Notepad2 or Notepad++

You can also just open the documents in their native application and ***save*** them as txt (UTF-8) files.

2) Copy these converted files to the **MMM/translation_input** folder

8.3. Run the *translate* script

To translate the documents run the ***translate*** script.

The files will be translated and stored in the **MMM/translation_output** folder

The names of the output files will be identical to those placed in the **translation/input** folder except for a suffix that is appended to them with the abbreviation of the target language and the indication that it is a Moses translation.

Example: If you input the file 100.pt you will get a translated 100.pt.en.moses file (if "en" is the abbreviation of the target language).

Once finished, you can move these files to another folder that you may create anywhere you want and repeat the operation with other document(s).

If you add other original document to translate and you forget to move/delete from the **translation_input** and **translation_output** folders documents already translated, when you run the ***translate*** script, only the new documents will be translated.

If you move/delete the previous MT translations from the ***translation_output*** folder, but you forget to move/delete also the original files — and you run the ***translate*** script — Moses will translate those documents again.

8.4. Purpose of MT translation

If you just need to have the content for gisting purposes without any formatting and/or post-editing, you can use the translation right away. It will be, as the converted source file, in txt format (UTF-8).

If you want to use MT output for translation purposes with a CAT tool, you can use the utility provided in MMM — **MOSES2TMX_1.032.EXE** — to convert the source file and the MT output file into a translation memory file (tmx).

And that's all!

9 — Scoring your MT translations

Script file: ***score-**** in the ***Moses-for-Mere-Mortals/scripts*** folder

In this Section we assume that you are familiar with SMT concepts or that you have read the Section on Automatic Evaluation in **What Makes Moses Tick.**

You may want to test Moses performance with documents you have already translated to have an idea of what to expect in terms of usefulness for translation purposes.

Therefore, you may use documents that were previously translated and score MT output using the human translation as reference.

9.1. Preparing for scoring

To score an MT translation, you must have 2 perfectly aligned files: a file in the source language and a file with a human translation in the target language.

You can score a single document or a batch of documents and MMM will produce scoring reports for each of the documents individually.

To obtain these perfectly aligned files, the surest way is to use a translation memory file (tmx) — if you use CAT tools for your translations — or have them aligned in tmx format.

When you have the tmx file:

1) Use the utility provided in MMM — **EXTRACT_TMX_CORPUS_1.043.EXE** — to separate that bilingual file into two monolingual files perfectly aligned.

2) Copy the original files to the **/MMM/translation_input** folder

3) Copy to the **/MMM/translation_reference** folder the file(s) containing the human translation of your document(s) with exactly the same name as the original files (but different language abbreviations of course), and with the extension **.ref**.

9.2. Running the translate and score scripts

1) Run the ***translate**** script.

   The files will be translated and stored in the ***MMM/translation_output*** folder

2) If you want, open the ***score*** script, and give a name to the scoring of this/these document(s) in the ***batch_user_note*** field.

This information will appear in the name of the scoring file for each document. This may be useful if you want to identify the scoring of MT output for further reference.

2) Then run the ***score*** script

The ***score*** script will just take the files that are in the ***MMM/translation_output*** and ***MMM/translation_reference*** and produce individual scores for each document.

The score report for each individual document (if there is more than one document) is stored in the ***/MMM/reports/scorer*** subfolder.

And that's all!

PART 3

– EXPLORING MOSES/MOSES FOR MERE MORTALS OPTIONS –

10 — Installing MMM on a non-default location

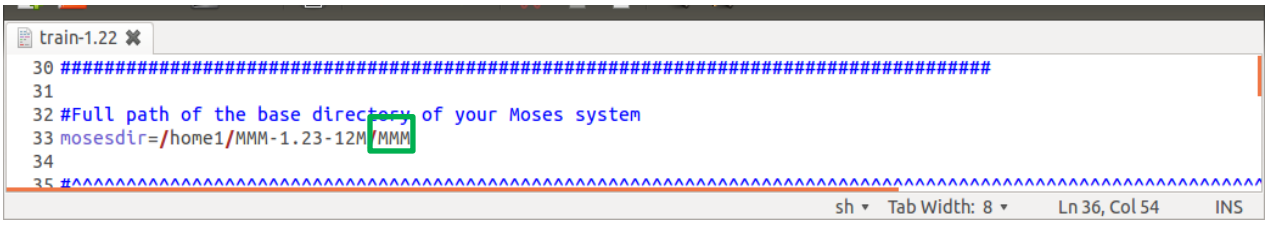
All the script files in the **Moses-for-Mere-Mortals/scripts** folder

With MMM, you may train as many corpora/language pairs as you want in a single MMM installation, but you may also want to have different installations, for instance, for different training corpora or different language pairs in order to do simultaneously translations of documents with different languages pairs (if your computer is powerful enough for that).

If you want to have an MMM installation in a non-default location, you will have to redefine the path that will be used by MMM in all the scripts.

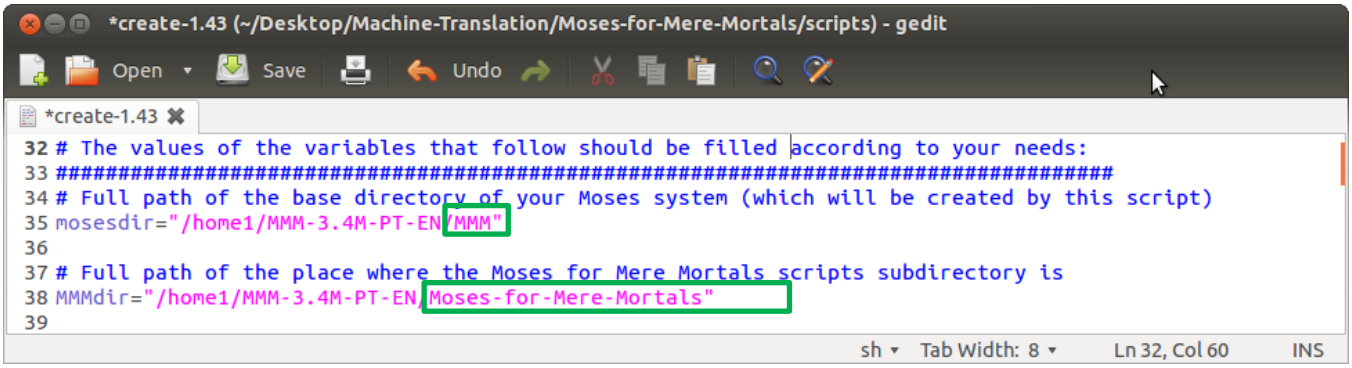
You just have to:

1) Create a folder — that can be called whatever you want — in the location you want;
2) Copy the uncompressed **Moses-for-Mere-Mortals*** folder you downloaded from the MMM website to that folder and rename it **Moses-for-Mere-Mortals**;
3) Open all the script files and — in the ***mosesdir*** field — replace the part of the path *$HOME/Desktop/Machine-Translation* by the name of the location where you want to install MMM, leaving the rest of the respective path unchanged (/MMM).



Screenshot 29 — Path to be changed in all the scripts. The final part of the path (/MMM) must not be changed!

4) In the **create** script, besides changing the path in the ***mosesdir*** field, you must also change the path in the ***MMMdir*** field by also replacing the part of the path ***$HOME/Desktop/Machine-Translation*** by the name of the location where you want to install MMM, leaving the rest of the respective path unchanged (/Moses-for-Mere-Mortals).



Screenshot 30 — Paths to be changed in the *create* script. The final parts of the path (*/MMM* and */Moses-for-Mere-Mortals*) must not be changed!

The example in the screenshots above refers to a new MMM installation created in a disk named **home1** to train a PT-EN engine with a 3.4M-PT-EN corpus:

mosesdir="/home1/MMM-3.4M-PT-EN/MMM"

MMMdir= ="/home1/MMM-3.4M-PT-EN/**Moses-for-Mere-Mortals"**

We suggest that you start by changing the paths in all the scripts at the same time so that afterwards you don't receive error messages.

11 — Transferring training(s) to another location in the same computer or in another computer

Script file: ***transfer-training-to-another-location-**** in the ***Moses-for-Mere-Mortals/scripts*** folder

This script should be used on the computer in which the training that is to be transferred was made (not on the target computer).

You may want to transfer the training in a MMM installation to another location in the same computer or to another computer, for instance, because you may want to share it with a colleague or do the training of a corpus — which is a more demanding operation — in a (more) powerful PC and afterwards transfer it to a laptop to do the translations.

You can do it with the ***transfer-training-to-another-location*** script. Take into consideration that, in this operation, you will transfer all the trainings that you have done in that particular MMM installation as — if you do several trainings, for instance a tuning of an already trained corpus — MMM will reuse the part that was already done in the first training.

So, if you are planning to use a particular training in another location/computer — if it is of a sizable corpus — it may be a good idea to do only one training in that MMM installation and therefore, when you use the ***transfer-training-to-another-location*** script, you will have only an individual training transferred.

We recommend that you transfer the training to a new installation in the new location.

Your original trainings are not affected by this operation (they are not erased, they are just copied).

To prepare the training to be converted and copied to another location:

1) Open the ***transfer-training-to-another-location*** script

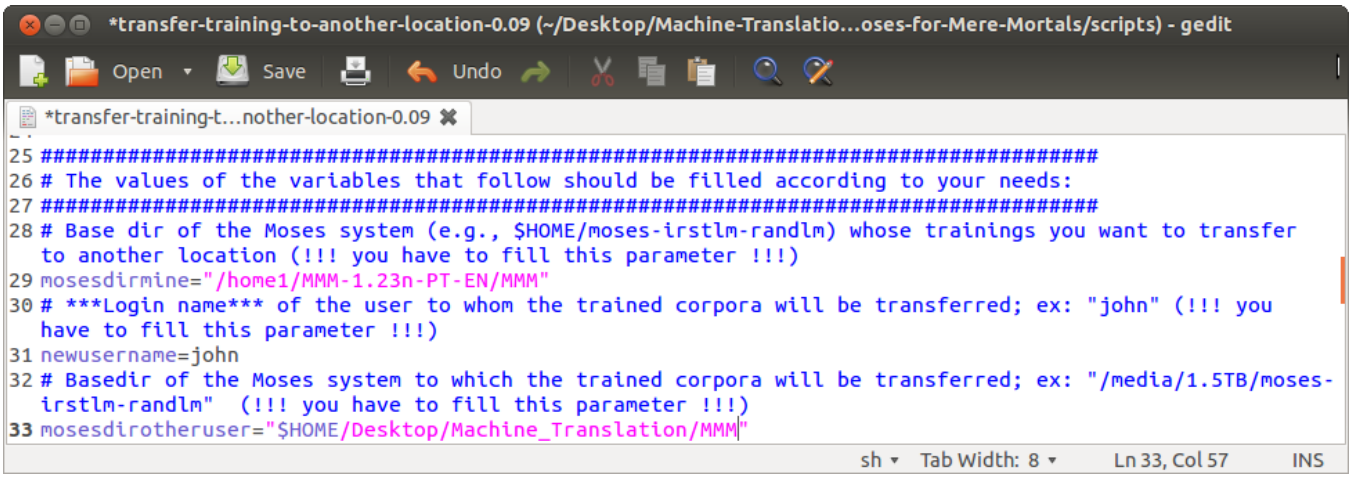
2) The field ***mosesdirmine*** is by default the (default) location of your MMM installation.

   If you have the MMM installation in a non-default location, fill in the name of the path, if you haven't done it yet.

3) Fill in the field ***newusername*** which must the login name in the computer to where you want to transfer the training.

4) Fill in the field ***mosesdirotheruser*** which must be the location to which the training will be transferred.

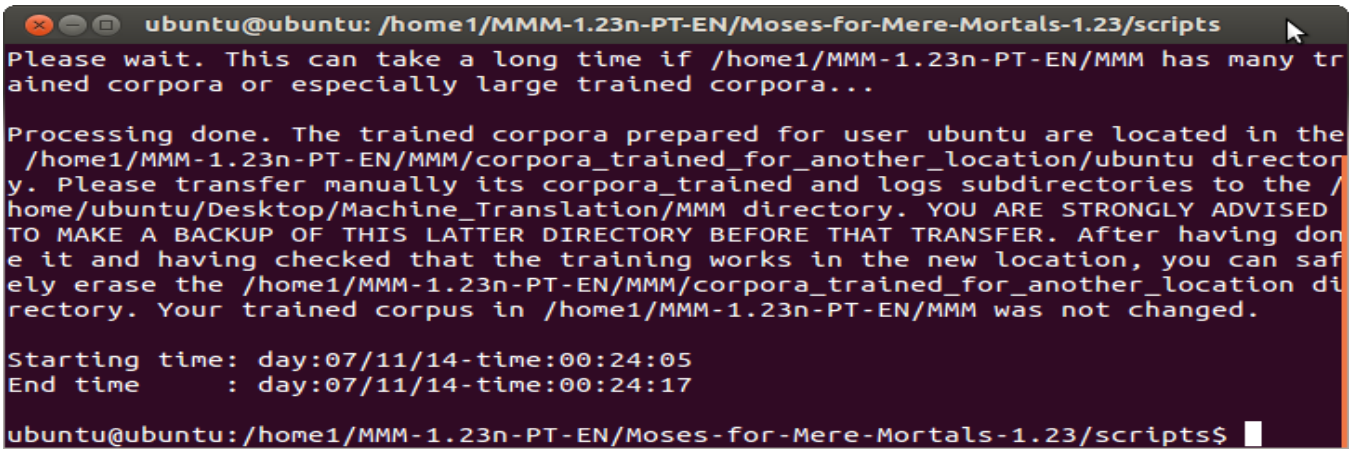
In the example below, the training was done in a disk called ***home1*** in an installation named ***MMM-1.23n-PT-EN***. The training is to be transferred to another computer, for which the username is 'ubuntu', to an installation named ***Machine_Translation*** in ***$HOME/Desktop/***.



Screenshot 31 — The *transfer-training-to-another-location* script

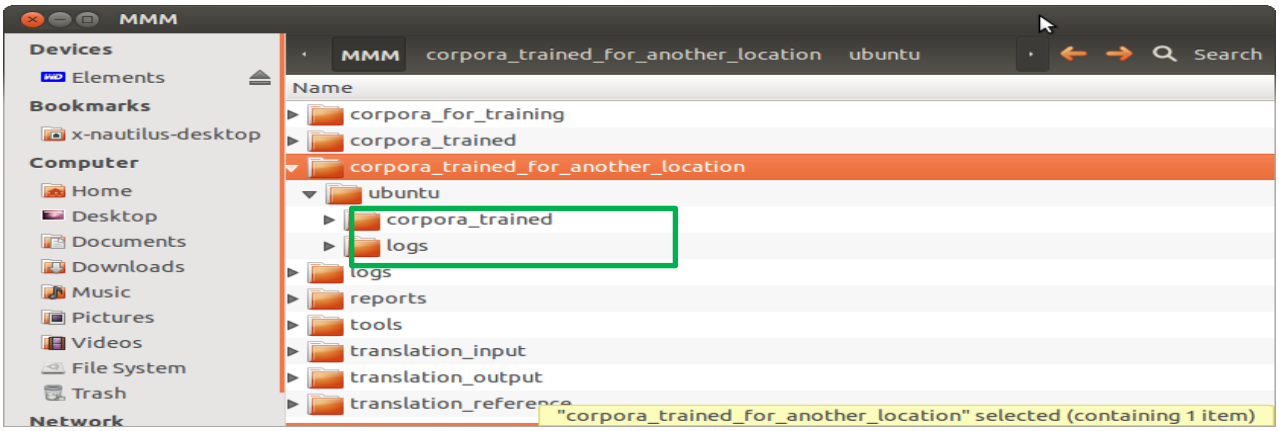
5) Run the script. It may take some time, depending on the size of the training(s) you want to transfer.

6) When the operation is complete, you see in the **Terminal** the message that the processing was done.



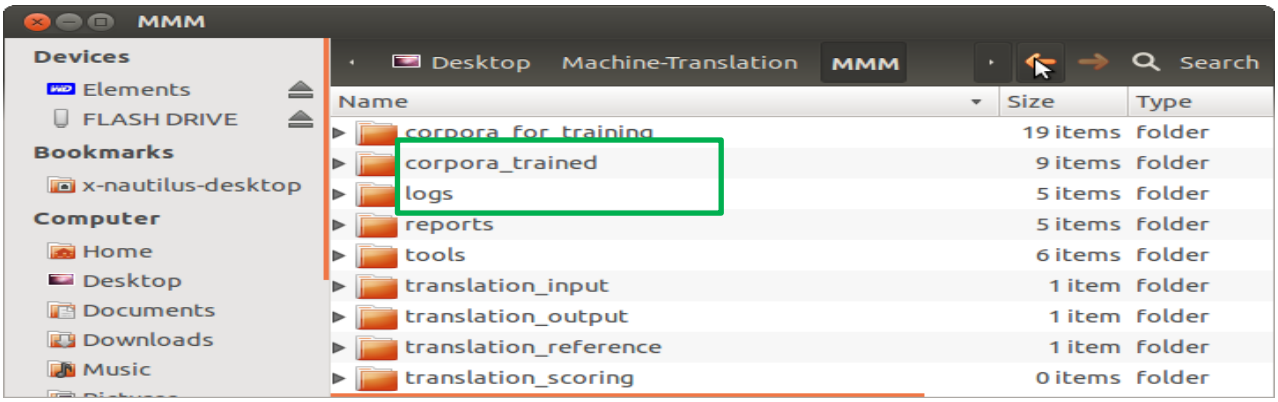
Screenshot 32 — Transfer training to another location — Messages in the Terminal

7) In the ***MMM*** folder, a new folder is created, with the training(s) ready to be transferred, named ***corpora_trained_for_another_location***.



Screenshot 33 — Corpora_trained_for_another_location folder.
The 2 folder to be copied to another location highlighted in green.

8) Copy these 2 folders to the new location in your computer or to the new computer (via an USB key or an external disk if it is a large corpus) where you want them to be used.



Screenshot 34 — Transferred corpora copied to the new location

If you want, you can copy the training(s) to an installation which already has other trainings. In that case you must merge — in the target location — the ***corpora_trained*** and ***logs*** folders with the converted trainings for transfer in the ***corpora_trained*** and ***logs*** folders.

9) After you have transferred the corpora to their intended location, you can safely erase the ***MMM*/corpora_trained_for_another_location** folder.

And that's all!

12 — *Train* and *translate* scripts — Exploring some interesting parameters

Script files: ***train**** in the **Moses-for-Mere-Mortals/*scripts*** folder
translate* in the **Moses-for-Mere-Mortals/*scripts*** folder

In this Section we assume that you are interested in exploring training and translation options in order to see if you can get better MT results. We also assume that you are familiar with SMT concepts or that you have read the Annex on **What Makes Moses Tick**.

In MMM, you can define 70 training parameters and 18 translation parameters… but if you want to explore them all you are no longer a "mere mortal" and you must refer to the comments in the script files and the Moses Guide.

Here we only highlight a few **interesting parameters** that you can easily change in MMM to see how it affects MT performance. Some options may require more computer resources than others, or more time for training, but the results may be better.

The MMM defaults are mostly the Moses defaults (for the whole list of defaults, see Annex 2). However we have in some cases different defaults for some parameters as we tried to strike a balance between the quality of MT output and speed and resource consumption.

But there is nothing like trying to find out!

In the following subsections we present a brief description of a few interesting parameters.

12.1. Interesting training parameters

These are parameters that you can easily change in MMM and that have (some) impact in the training of a corpus:

1) ***lngmdl*** and ***smoothing*** and ***gram***
2) ***tuning***

12.1.1. lngmdl, smoothing and gram

The Language Model is very important for training, in terms of MT output quality and time to train.

In MMM you can choose between 2 Language Models by changing the parameter ***lngmdl***:

- IRSTLM (1)
- RandLM (5)

If you use IRSTLM, you can also choose the smoothing parameter ("***s***"). Possible values are ***witten-bell***; ***kneser-ney***, ***improved-kneser-ney***. The default is ***improved-kneser-ney***.

You can train both of them with different n-grams (parameter ***Gram***).

The n-gram order can very significantly influence the results. The higher the better, but it comes at a price in terms of memory and time required.

You can use values between 3 and 9. The default is 7 grams

12.1.2. tuning

Tuning is a phase that can take quite some time. Typically it leads to better results.

You may want to train a corpus without tuning and afterwards do the tuning. As MMM reuses the previous training without tuning, it will only do the tuning part of the training.

Furthermore, you cannot easily estimate the duration of tuning beforehand, since the number of its runs is highly variable.

The user can control the number of tuning runs (iterations) through the parameter **maxruns**.

Possible values for tuning: ***no-tuning***, ***tuning*** and also ***apply_previous_weights***.

Default: no-tuning.

12.1.2.1. Tuning approach

Since tuning can be a very long phase and since its only useful — and very important — product is a set of weights that it transfers to the ***moses.ini*** file, you could perhaps invest in a single long tuning for each pair of languages that you are interested in and copy the weights from such a **moses.ini** to every other **moses.ini** that will be created for that same language pair, a very big time saving trick. This works as long as the kind of content does not change much.

If the files used for tuning are representative of the types of documents you are interested in, they should lead to better results than the default values used when no tuning is done.

In practice, you can first train a corpus without tuning (and that is the default in MMM), translate a representative text and then score that translation with the ***score*** script or see the score result of the test file you used for the training test.

Then, you can retrain the same corpus with tuning and check the score of the same text or test file.

As the ***train*** script reuses the previously non-tuned training of the same corpus with the same parameters, you will just do a new tuning and a new training test if no other parameters are changed.

You can repeat this for several representative texts. If the scores obtained with tuning are significantly higher than those obtained without tuning, then you can use the tuning weights for all the similar corpora of that language pair.

You can also test the 3 tuners available in MMM. They have different strengths and weaknesses. Nothing like trying to see where you obtain best results.

12.1.2.2. no-tuning

With **no-tuning,** the corpus will be trained without tuning, which means that the training will take less time, but also that the system will not try to automatically find the best possible weights for the several models (the ***weight_l***, ***weight_d***, ***weight_w*** and ***weight_t***)

12.1.2.3. tuning

With ***tuning*****,** MMM will use the tuner you have defined (or the default) to try to find the best scores for the test corpus you have provided which … hopefully … will improve MT output.

12.1.2.4. tuning_type

In MMM, you can do the tuning with 3 tuners: **mert**, **kbmira**, **pro**. The default for the **tuning_type** is **mert**.

If you want to retune the same corpus with the 3 tuners, you will generate different trainings when you change this parameter.

As the tuning is a random process and it gives different results every time you run it — even with the same tuner and the same number of runs. Take into consideration that, if you run the tuning a second time without changing the tuning parameters, the previous weights will be replaced.

12.1.2.5. maxruns

As tuning is a very intensive task, you may want to control its duration.

Possible values: positive values; only use values >=1. Number of runs you define. The default is: 25

Depending on the tuner, it may do all the 25 runs or it may stop (well) before that when it reaches a point that it considers that the weights are as optimised as possible.

12.1.2.6. apply_previous_weights

With the option ***apply-previous-weights***, MMM will use the ***weight.ini*** file that is in the **MMM/corpora_for_training** folder. There is already a **weight.ini** file in this folder and MMM will use it by default.

If you want you can change it but it must have the same structure of the file already there.

This file is by default defined in the parameter **previous_weight_file**="weight.ini".

To be on the safe side, you can do a rename of the weight.ini file in the **MMM/corpora_for_training** before replacing it by another. You may still want to use it latter!

12.2. Interesting translating parameters

Here we present a brief description of some interesting parameters that you can easily define to see if they improve Moses output in the translation stage:

1) ***weight_l***, ***weight_d*** and ***weight_w***
2) ***monotoneatpunctuation***
3) ***searchalgorithm***, ***cubepruningpoplimit***, ***stack***
4) ***distortionlimit***

12.2.1. weight_l, weight_d and weight_w

These parameters can only be used for a non-tuned corpus. For trainings with tuning, the tuning weights (in moses.ini prevail… as that is exactly the purpose of tuning!) and these parameters are deactivated.

Depending on whether you are using MT output for gisting purposes or for translation, you may prefer to have a more accurate output, although less fluent, or a more fluent output although not so accurate.

Also depending on your target language syntactic structures, you may want to limit or extend the range for the reordering model to reorder words in a segment.

So you may want to play with these parameters in the ***translate*** script and see how they affect MT output.

Weights for language model (good values: 0.1-1; default: 1); ensures that output is fluent in target language

weight_l=1

Weights for reordering model (good values: 0.1-1; default: 0.5); allows reordering

weight_d=0.5

Weights for word penalty (good values: -3 to 3; default: 0; negative values favour large output; positive values favour short output); ensures translations do not get too long or too short

weight_w=0

12.2.2. monotoneatpunctuation

With this parameter activated, Moses adds walls around punctuation ,.!?:;". Specifying reordering constraints around punctuation is often a good idea.

Option: 1= Do; Any other value = do not. Default: 0.

12.2.3. searchalgorithm, cubepruningpoplimit, stack

Especially with very large trained corpora (with several million segments), translation can be slow.

According to the Moses Manual, to get faster performance than the default Moses setting at roughly the same performance, use the parameters **searchalgorithm**=1, **cubepruningpoplimit**=2000 and **stack**=2000.

In MMM, these are the defaults as in our tests it seemed not to have a negative impact in Moses performance and it saves time.

With cube pruning, the size of the stack has little impact on performance, so it should be set rather high. The speed/quality trade-off is mostly regulated by the -cube-pruning-pop-limit, i.e. the number of hypotheses added to each stack

You may also try to reduce both of the latter 2 parameters to values of 500 or less (say, 100) and experiment to determine if they significantly change the translation quality.

1.2.2.4. distortionlimit

In this parameter, you can define the level of reordering allowed during translation, i.e. the maximum number of words that can be skipped. The default is 6.

Depending on the syntactic structure of the source and target language, namely on how close or how distant they are, you may want to increase this value. So, it might be interesting to test this parameter for some language pairs like German.

Long distance reordering is a very active area of research and there are sophisticated techniques that try to deal with this issue.

If you want to see how it will affect MT quality, take into consideration that large-scale reordering is often arbitrary and instead of improving MT output quality, may worsen it.

In general, limiting reordering allows to speed up the translation process and may also improve MT quality.

ANNEX 1. What Makes Moses Tick

This section is meant for absolute beginners. If you are already familiar with Statistical Machine Translation, skip it.

Here we present, in extremely simplistic terms, a general idea of what Moses does so that the several stages of using Moses via Moses for Mere Mortals are better understood.

In this overview, we use illustrations in presentations by Philipp Koehn available on the Internet and screenshots from Moses for Mere Mortals.

1.1. Statistical Machine Translation (SMT)

SMT was a breakthrough in the field of Machine Translation with the publication of the seminal paper ***Statistical Phrase-Based Translation*** by Philipp Koehn, Franz Josef Och and Daniel Marcu in 2003[24].

In 2007, the open-source Moses SMT system was first made available within EuroMatrix, a research project co-funded by the European Commission.

As SMT is language-independent — i.e. it is possible to train whatever language pair without the long and consuming work demanded by rule-based MT — it soon became the basis for services like Google or Bing.

The breakthrough was that it builds on the work done by the IBM Labs in the 90's, evolving from word-based (Figure 1) to phrase-based models (Figure 2) … and that made all the difference!

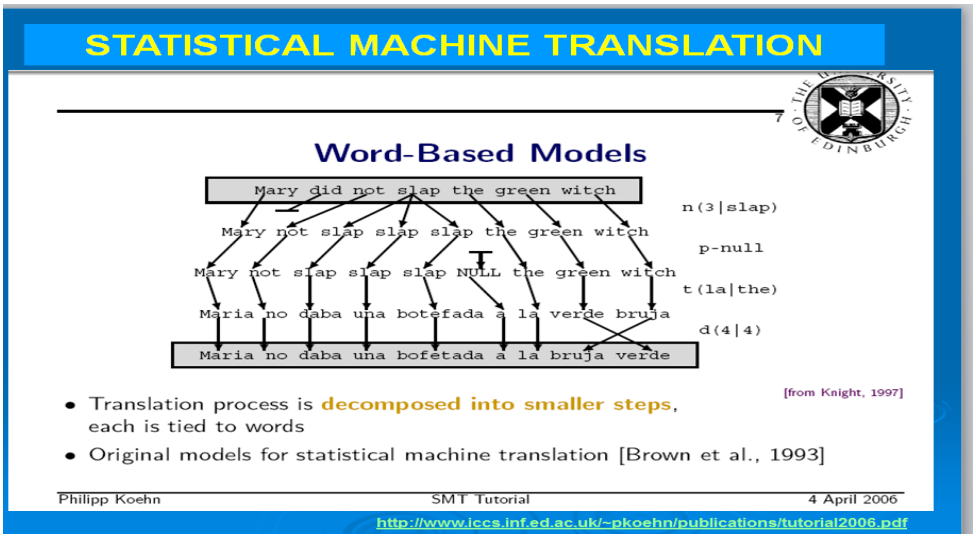


Figure 1 — Word-based Models

24 http://homepages.inf.ed.ac.uk/pkoehn/publications/phrase2003.pdf

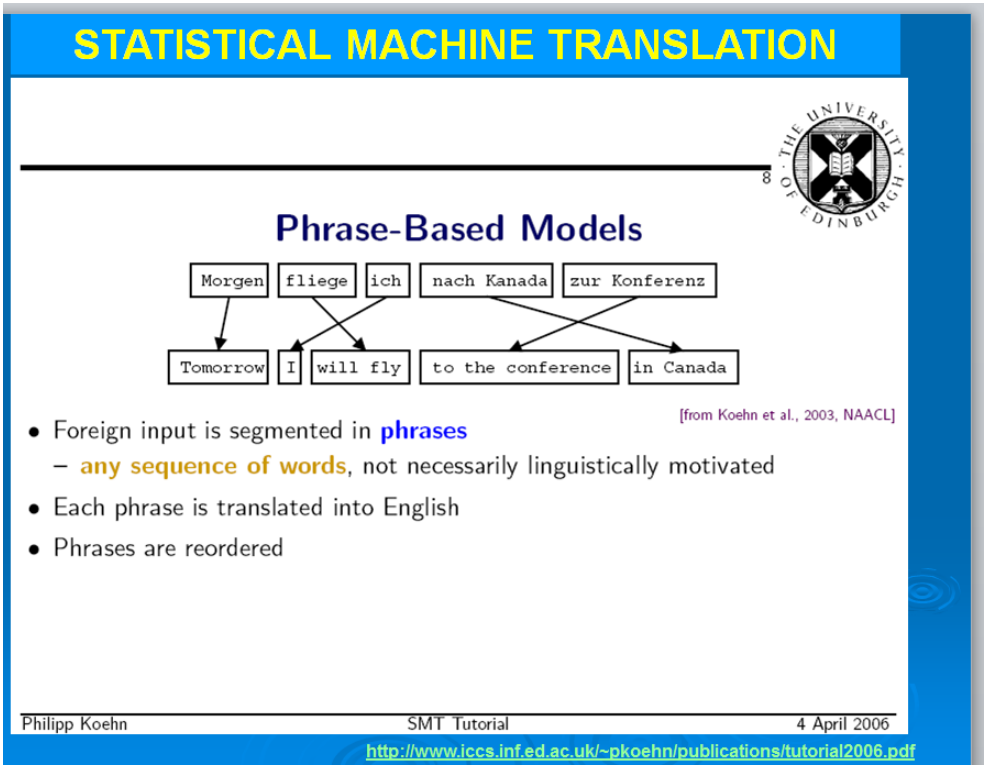


Figure 2 — Phrase-based Models

Very important also is that it became recently viable to have large corpora using a "by-product" of MT research: the Translation Memories.

It is the training of these large corpora built from high-quality human translations — aligned as translation memories and later converted — that made possible Statistical Machine Translation.

Therefore, there is in fact a *synergy between human translation and machine translation*.

For professional translators, this can be a "virtuous circle": machine translation is enriched with human translations (that are relevant for a particular domain) and, in turn, machine translation helps translators in their work.

1.2. What is a bilingual corpus

Parallel data is a collection of sentences in two different languages, which is **sentence-aligned**, in that each sentence in one language is matched with its corresponding translated sentence in the other language. It is also known as a **bitext**.

To train MT engines with Moses you need (a large amount of) data.

Now with the widespread use of CAT tools, translated documents are usually stored as translation memory files.

Translation memories are bilingual files which contain the source segments and the target human perfectly aligned translation in Translation Memory eXchange format (tmx) files (Figure 3).

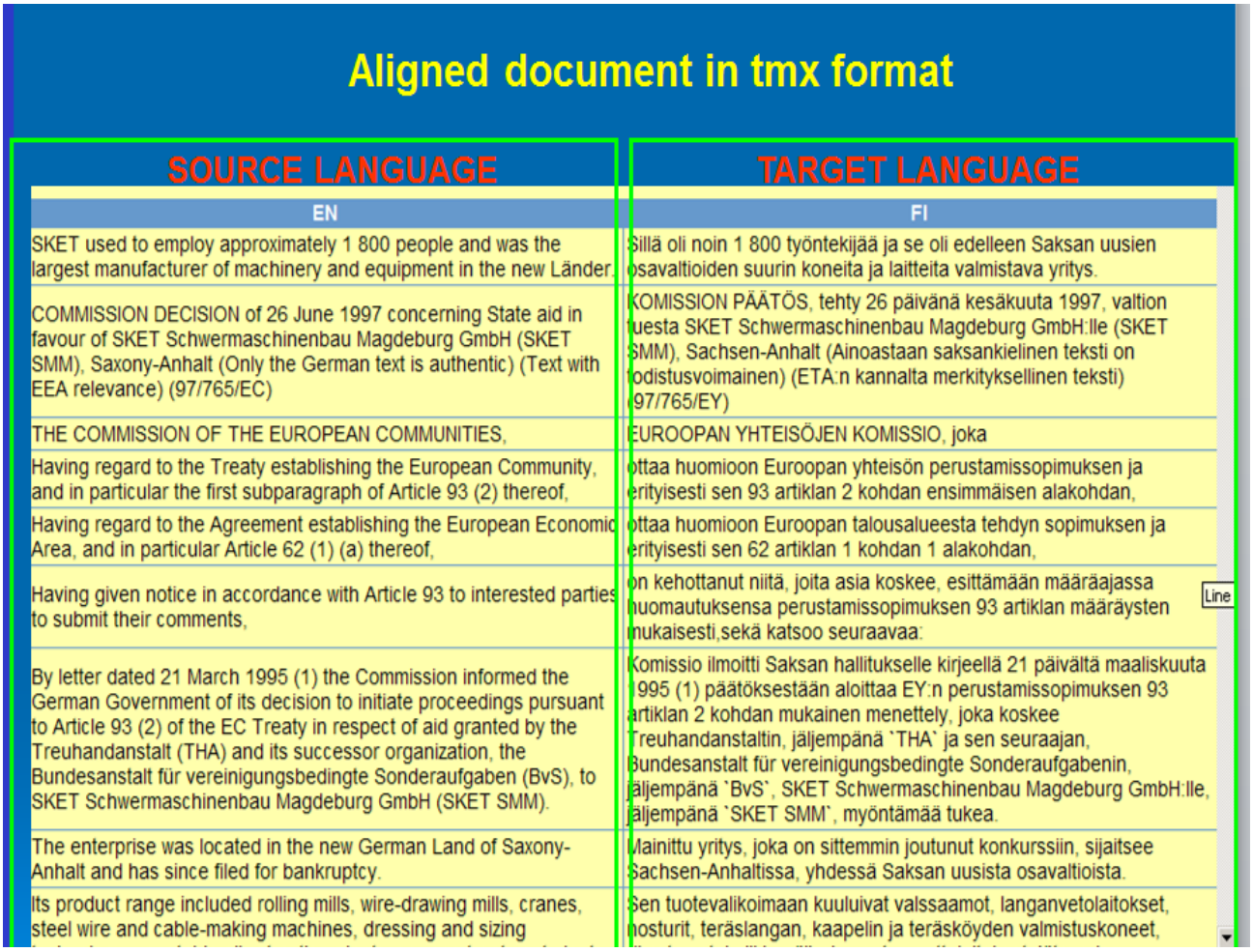


Figure 3 — Display of a tmx file

These tmx files can be merged and subsequently split into 2 files with the application **EXTRACT_TMX_CORPUS_1.043.EXE** included in MMM.

The resulting 2 files — one with the source segments and the other with the target segments — are perfectly aligned in text only (UTF-8) format (i.e., stripped of formatting). This is what a corpus looks like (Figure 4).

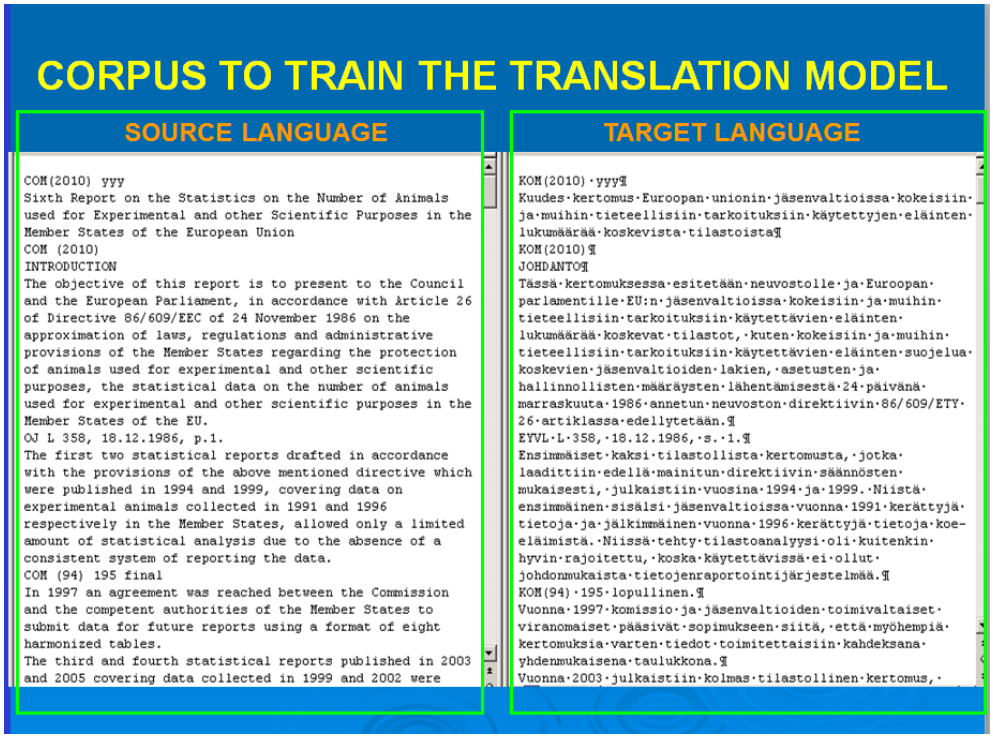


Figure 4 — Bilingual corpus in UTF-8 format

Moses will process those files, in what is called as a 'training' and will create what is called 'MT engines'. When the training is finished you will be able to start translating with it.

There are several corpora freely available which allow building MT engines for all the EU official languages and also for some other languages like Chinese, Arabic.

You can add your domain-specific data to these corpora if you have it. If you don't have your documents aligned you can use open-source tools like hunalign (http://mokk.bme.hu/resources/hunalign/) or LFAligner (http://sourceforge.net/projects/aligner/).

Bilingual corpora can also be aligned at paragraph level, but it is generally thought that it is better to have them aligned at sentence (segment) level for MT purposes.

1.3. Tokens and n-grams

Important concepts in SMT are tokens and n-gram units.

Tokens are the basic unit in a machine translation process: tokens are a sequence of characters, such as words, punctuation or symbols, separated by a space. When the training is launched, the corpora are first tokenised.

An **n-gram** is a subsequence of n number of (1, 2, 3, etc) items in a larger sequence.

In a Language Model, n-grams are sequences of tokens. In MMM, you can choose the number of n-grams you want to use in a training. The default is 7, but you can use values between 3 and 9. Obviously, 7-gram Language Models take more time than 3-gram models and generate bigger files.

In Phrase Tables and Reordering Tables, n-grams are sequences of pairs of source and target language tokens.

In MMM, the default is 7, but you can choose values between 3 and 9. As for the Language Model, the higher the n-grams, the larger the Phrase Table will be, the more time it will take to generate the Phrase Table and the bigger the files will be.

Depending on what you are using MT for, you may prefer speed to quality or vice-versa.

1.4. What is a monolingual corpus

To train MT engines with Moses you also need monolingual data for certain parts of the training to generate:

i) the Language Model, which tries to ensure the fluency of the machine translation output

ii) the Recaser Model so that the machine translation output has the right case in words that have capital letters.

You can use the target language file of the bilingual corpus already extracted (see Figure 5). But you can also add to it texts in the target language of the language pair you want to train.

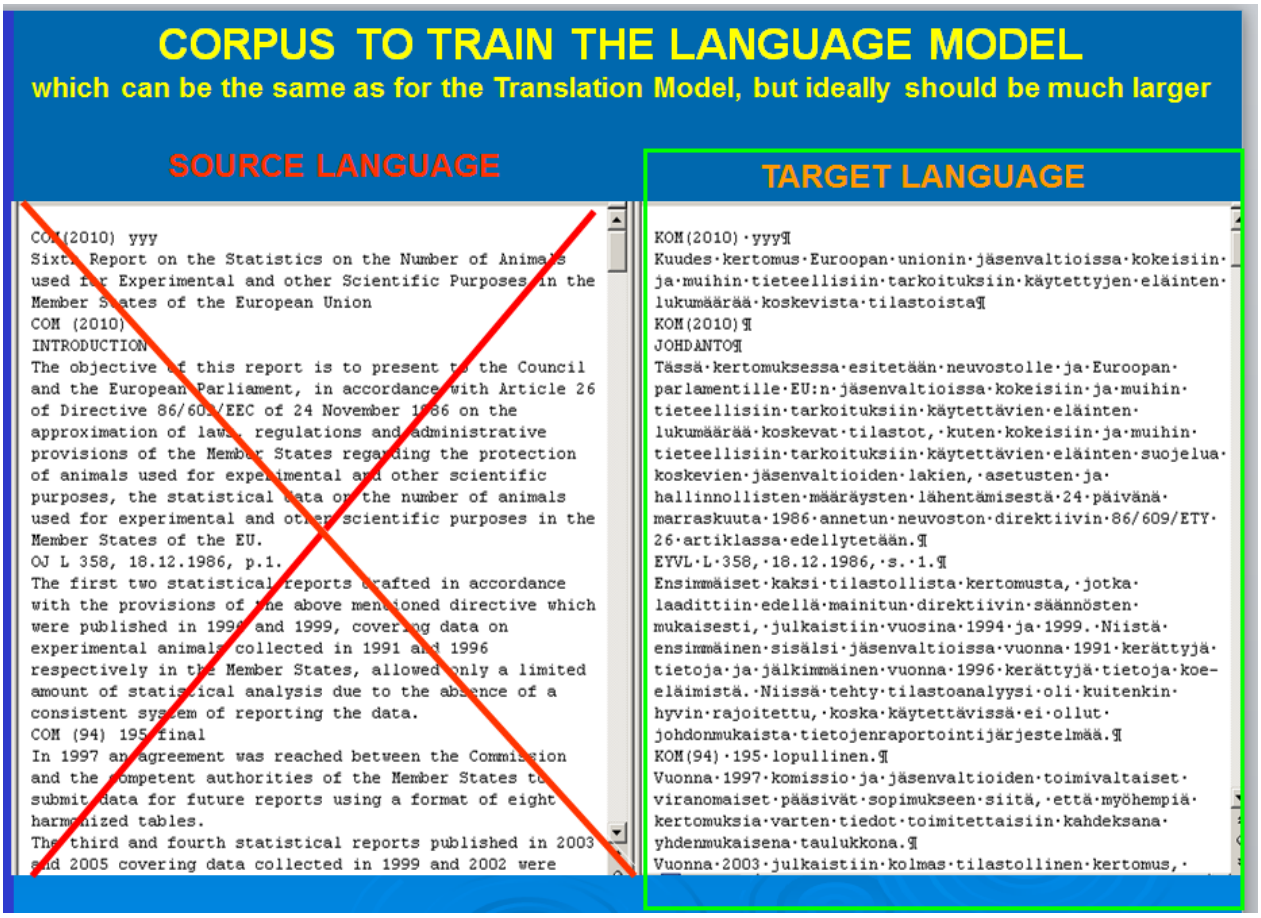


Figure 5 — Monolingual corpus in UTF-8 format

Monolingual data is much easier to get just by converting documents into the txt (UTF-8) format to be used by Moses to generate the language model.

1.5. What is training a corpus

When you have the corpora, these must be processed by MMM/Moses. The steps involved may differ depending on the options concerning language model, tuning, etc.. However, in general terms and just to have an idea, these are the steps in a training with the IRSTLM language model:

1) **Corpus preparation**:
    a. Tokenisation: insertion of spaces between (e.g.) words and punctuation.
    b. Lowercase data.
    c. Cleaning: long sentences and empty sentences are removed as they can cause problems.
    d. Furthermore, some characters that have proven to give problems are eliminated or converted.
2) A **language model** (LM) is generated using one of the LM tools — IRSTLM (by default in MMM) or RANDLM;
3) **Word alignment**: The training corpus — which is already aligned at segment level — is further aligned at subsegment level using a word aligner.

    In MMM, the word aligner is MGIZA++;
4) A **translation model** (phrase table) is generated which contains phrase-to-phrase translations extracted from the training corpus;
5) Other models are also generated like the **Reordering and Recasing Models**;
6) Optionally, **tuning** can also be done to try to improve the quality of MT output;
7) Optionally, it can also automatically **score** MT output with automatic metrics.

    In MMM, are available the BLEU and NIST metrics.

```
 2 ================================================================================
 3 *** Duration ***:
 4 ================================================================================
 5 Start time:                                  day:30/10/2014-time:21:33:16
 6 Start language model building:               day:30/10/14-time:21:33:16
 7 Start recaser training:                      day:30/10/14-time:21:41:31
 8 Start corpus training:                       day:30/10/14-time:21:43:11
 9 Start memory-mapping:                        day:30/10/14-time:23:11:49
10 Start tuning:
11 Start test:                                  day:30/10/14-time:23:18:15
12 Start scoring:                               day:30/10/14-time:23:21:28
13 End time:                                    day:30/10/14-time:23:21:30
14 ================================================================================
15 *** Languages*** :
16 ================================================================================
17 Source language: pt
18 Target language: en
19 ================================================================================
20 *** Training steps in fact executed *** :
21 ================================================================================
22 Language model building executed=yes
23 Recaser training executed=yes
24 Corpus training executed=yes
25 Parallel training executed=yes
26 First training step=
27 Last training step=9
28 Corpus memmapping executed=yes
29 Tuning executed=no
30 Training test executed=yes
31 Scoring executed=yes
```

Figure 6 — Example of the report of a training of the 200 000 Demo corpus (PT-EN) with the MMM defaults (without tuning)

1.6. What a Translation Model (Phrase Table) looks like

The phrase table is a kind of bilingual dictionary with probabilities computed during the training process. The phrase table is what is used by the system to try to guarantee the correctness of the translation, i.e. that “the black cat” is translated in the target language by its equivalent (“le chat noir”) and not by something like “le chat jaune”.

Just to have an idea, a 3.4 million segment corpus (with 64.4 million words (EN)), which is available in the MMM website, can generate a phrase table with about 200 000 entries ... with different degrees of reliability (probabilities as computed during the training).

Depending on the choice made, the Phrase Table may have entries with up to 9 grams. In MMM, the default is 7.

```
5192 actividades de investigação e ||| the activities of research and ||| 1 0.136191 0.5 0.0595533 2.718 ||| 0-1 1-2 2-3 3-4 ||| 1 2 1
5193 actividades de investigação ||| activities of research ||| 1 0.154862 0.5 0.225131 2.718 ||| 0-0 1-1 2-2 ||| 1 2 1
5194 actividades de investigação ||| the activities of research ||| 1 0.154862 0.5 0.0641159 2.718 ||| 0-1 1-2 2-3 ||| 1 2 1
5195 actividades de ||| activities of ||| 0.2 0.154862 0.5 0.225131 2.718 ||| 0-0 1-1 ||| 5 2 1
5196 actividades de ||| the activities of ||| 1 0.154862 0.5 0.0641159 2.718 ||| 0-1 1-2 ||| 1 2 1
5197 actividades definidas no artigo 1º ||| activities as set out in article 1 ||| 1 0.000625078 1 0.00279051 2.718 ||| 0-0 1-1 1-2 1-3 2-4 3-5 4-6 ||| 1 1 1
5198 actividades definidas no artigo ||| activities as set out in article ||| 1 0.00546943 1 0.00279051 2.718 ||| 0-0 1-1 1-2 1-3 2-4 3-5 ||| 1 1 1
5199 actividades definidas no ||| activities as set out in ||| 1 0.00546943 1 0.00294 2.718 ||| 0-0 1-1 1-2 1-3 2-4 ||| 1 1 1
5200 actividades definidas ||| activities as set out ||| 1 0.0252264 1 0.006 2.718 ||| 0-0 1-1 1-2 1-3 ||| 1 1 1
5201 actividades no âmbito do presente ||| activities under this ||| 0.5 0.00212585 1 0.0723819 2.718 ||| 0-0 1-0 2-0 2-1 3-1 4-2 ||| 2 1 1
5202 actividades no âmbito do ||| activities under ||| 0.5 0.00340136 1 0.076518 2.718 ||| 0-0 1-0 2-0 2-1 3-1 ||| 2 1 1
5203 actividades ordinárias ( seja antes ou após ||| ordinary activities ( either before or after ||| 1 0.0272125 1 0.00705941 2.718 ||| 1-0 0-1 2-2 3-3 4-4 5-5 6-6 ||| 1 1 1
5204 actividades ordinárias ( seja antes ou ||| ordinary activities ( either before or ||| 1 0.0395817 1 0.00834295 2.718 ||| 1-0 0-1 2-2 3-3 4-4 5-5 ||| 1 1 1
5205 actividades ordinárias ( seja antes ||| ordinary activities ( either before ||| 1 0.0468384 1 0.00987248 2.718 ||| 1-0 0-1 2-2 3-3 4-4 ||| 1 1 1
5206 actividades ordinárias ( seja ||| ordinary activities ( either ||| 1 0.0702576 1 0.0542987 2.718 ||| 1-0 0-1 2-2 3-3 ||| 1 1 1
5207 actividades ordinárias ( ||| ordinary activities ( ||| 1 0.0702576 1 0.705882 2.718 ||| 1-0 0-1 2-2 ||| 1 1 1
5208 actividades ordinárias ||| ordinary activities ||| 1 0.107143 1 0.75 2.718 ||| 1-0 0-1 ||| 1 1 1
5209 actividades ou acontecimentos significativos desde o fim ||| activities or events since the end ||| 1 0.00278081 1 0.039504 2.718 ||| 0-0 1-1 2-2 3-2 4-3 5-4 6-5 ||| 1 1 1
5210 actividades ou acontecimentos significativos desde o ||| activities or events since the ||| 1 0.0139041 1 0.276528 2.718 ||| 0-0 1-1 2-2 3-2 4-3 5-4 ||| 1 1 1
5211 actividades ou acontecimentos significativos desde ||| activities or events since ||| 1 0.0905432 1 0.422535 2.718 ||| 0-0 1-1 2-2 3-2 4-3 ||| 1 1 1
5212 actividades ou acontecimentos significativos ||| activities or events ||| 1 0.0905432 1 0.633803 2.718 ||| 0-0 1-1 2-2 3-2 ||| 1 1 1
5213 actividades ou ||| activities or ||| 1 0.362173 1 0.633803 2.718 ||| 0-0 1-1 ||| 1 1 1
5214 actividades ||| activities of ||| 0.2 0.428571 0.142857 0.0241624 2.718 ||| 0-0 ||| 5 7 1
5215 actividades ||| activities ||| 0.625 0.428571 0.714286 0.75 2.718 ||| 0-0 ||| 8 7 5
5216 actividades ||| the activities ||| 1 0.428571 0.142857 0.213595 2.718 ||| 0-1 ||| 1 7 1
5217 activo de caixa ou seu equivalente que ||| cash or a cash equivalent asset which ||| 1 0.000226571 1 0.000902693 2.718 ||| 2-0 3-1 0-2 5-3 4-4 5-4 5-5 6-6 ||| 1 1 1
5218 activo de caixa ou seu equivalente ||| cash or a cash equivalent asset ||| 0.5 0.000372746 1 0.00442611 2.718 ||| 2-0 3-1 0-2 5-3 4-4 5-4 5-5 ||| 2 1 1
5219 activo de caixa ou ||| cash or a ||| 0.5 0.00216433 1 0.187793 2.718 ||| 2-0 3-1 0-2 ||| 2 1 1
5220 activo de ||| a ||| 0.010989 0.00320141 1 0.333333 2.718 ||| 0-0 ||| 91 1 1
5221 activo nas demonstrações financeiras da empresa que ||| asset in the financial statements of ||| 0.333333 3.05195e-08 1 0.020889 2.718 ||| 0-0 1-1 1-2 3-3 2-4 4-5 ||| 3 1 1
5222 activo nas demonstrações financeiras da empresa ||| asset in the financial statements of ||| 0.333333 8.73875e-07 1 0.020889 2.718 ||| 0-0 1-1 1-2 3-3 2-4 4-5 ||| 3 1 1
5223 activo nas demonstrações financeiras da ||| asset in the financial statements of ||| 0.333333 0.000750688 1 0.020889 2.718 ||| 0-0 1-1 1-2 3-3 2-4 4-5 ||| 3 1 1
5224 activo nas demonstrações financeiras ||| asset in the financial statements ||| 1 0.00476437 1 0.0520833 2.718 ||| 0-0 1-1 1-2 3-3 2-4 ||| 1 1 1
5225 activo nas ||| asset in the ||| 1 0.0110601 1 0.0520833 2.718 ||| 0-0 1-1 1-2 ||| 1 1 1
5226 activo numa base sistemática durante a ||| asset on a systematic basis ||| 0.333333 4.99335e-08 1 0.00222222 2.718 ||| 0-0 1-1 2-2 3-3 2-4 ||| 3 1 1
5227 activo numa base sistemática durante ||| asset on a systematic basis ||| 0.333333 1.04617e-06 1 0.00222222 2.718 ||| 0-0 1-1 2-2 3-3 2-4 ||| 3 1 1
5228 activo numa base sistemática ||| asset on a systematic basis ||| 0.333333 0.000898692 1 0.00222222 2.718 ||| 0-0 1-1 2-2 3-3 2-4 ||| 3 1 1
5229 activo numa ||| asset on ||| 1 0.00784313 1 0.111111 2.718 ||| 0-0 1-1 ||| 1 1 1
5230 activo ||| a ||| 0.010989 0.0104167 0.333333 0.333333 2.718 ||| 0-0 ||| 91 3 1
5231 activo ||| asset ||| 1 0.666667 0.666667 0.666667 2.718 ||| 0-0 ||| 2 3 2
5232 activos e passivos a serem alienados ||| assets and liabilities to be disposed of ||| 1 0.00276836 0.5 0.00164253 2.718 ||| 0-0 1-1 2-2 3-3 4-4 5-5 ||| 1 2 1
```

Figure 7 — Extract of a phrase table of a 7-gram PT-EN training showing entries from 1 to 7 gram and the computed probabilities.

1.7. What a Language Model looks like

The Language Model calculates the probability of a word after a given sentence or the probability of a given sentence and is generated during the training process. Its tries to guarantee the fluency of the MT output, i.e. that “Ich gehe nach Haus” is translated in the target language by “I am going home” and not by “I am going house” or something like that.

Depending on the choice made, the Language Model may have entries with up to 9-grams, that is, a sequence of 9 tokens (words, punctuation marks or symbols separated by a space). In MMM, the default is 7-grams.

Just to have an idea, the same 3.4 million segment corpus can generate a 7-gram language model (vocabulary) with about 155 million entries.

```
267 iARPA
268 loadtxt_ram()
269 1-grams: reading 276799 entries
270 done level1
271 2-grams: reading 3385018 entries
272 done level2
273 3-grams: reading 12891021 entries
274 ..done level3
275 4-grams: reading 24366544 entries
276 ....done level4
277 5-grams: reading 33019964 entries
278 ......done level5
279 6-grams: reading 38648425 entries
280 .......done level6
281
282 7-grams: reading 42413877 entries
```

Figure 8 — Number of entries in a 7-gram Language Model of the 3.4 million segment corpus PT-EN

This is what a language model looks like:

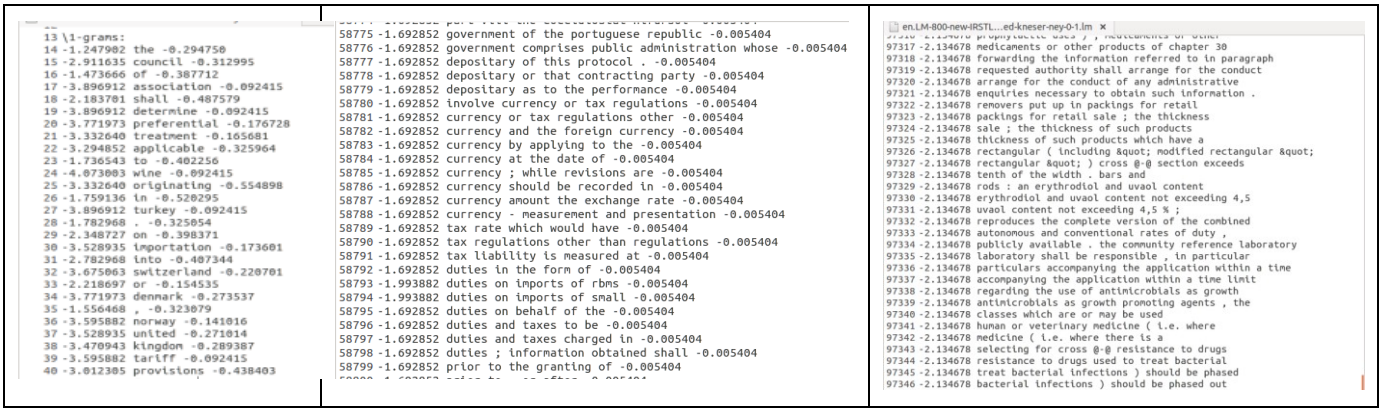


Figure 9 — Entries in a Language Model from 1-gram to 7-gram (here shown extracts with 1, 3 and 7 n-grams)

Moses supports several different language model toolkits (SRILM, KenLM, IRSTLM, RandLM). In MMM, are available the IRSTLM and RANDLM language models.

KenLM is not available in this version of MMM and SRILM is not automatically downloaded and installed as it requires a licence for non-academic purposes.

Training a Language Model from large/huge amounts of data can be memory and time expensive.

The IRSTLM features algorithms and data structures suitable to estimate, store, and access very large LMs.

RANDLM can be used to train really large LMs. It takes a very different approach to IRSTLM. It represents LMs using a randomized data structure. This can result in LMs that are ten times smaller than those created using the IRSTLM, but the quality of MT output can be lower, at least in our experience.

1.8. Reordering and Recaser Models

The Reordering Model will be used in the decoding (translation) process to try to get the words in a sentence translated in the correct order. For instance, to have the “the black cat” translated as “le chat noir” in French and not as “le noir chat”.

Before the training, all words in the training corpus are lowercased. The Recaser Model is used to recase words so that, for instance, “Union Européenne” is not translated as “european union”.

1.9. Tuning

This is a process that can be done at the end of the training of a corpus and which aims to improve the quality of MT output.

By translating a small tuning corpus (usually about 1000 to 2000 segments) repeatedly, the system will try to find the best weights between the components of a training (Phrase Table, Language Model and the other models) to achieve the best quality.

MMM automatically installs 3 tuners: mert (MMM default), pro and kbmira and you can define the number of runs that will be done.

A 'run' is the process by which Moses — the Decoder — translates the tuning corpus and afterwards scores it to see if it gets a better result. This operation is repeated as many times as defined in the training script.

In MMM, the default is 25 runs, i.e., in the tuning stage, the tuning corpus will be translated up to 25 times, depending on the tuner used.

This process takes quite some time and there is no assurance that the quality of MT output will be better than without it.

1.10. The translation process

Moses is the Decoder (the "translator") which translates new sentences by finding the highest scoring sentence in the target language in terms of exactness (according to the Translation Model) and fluency (according to the Language Model) from a list of candidate translations (the n-best list).

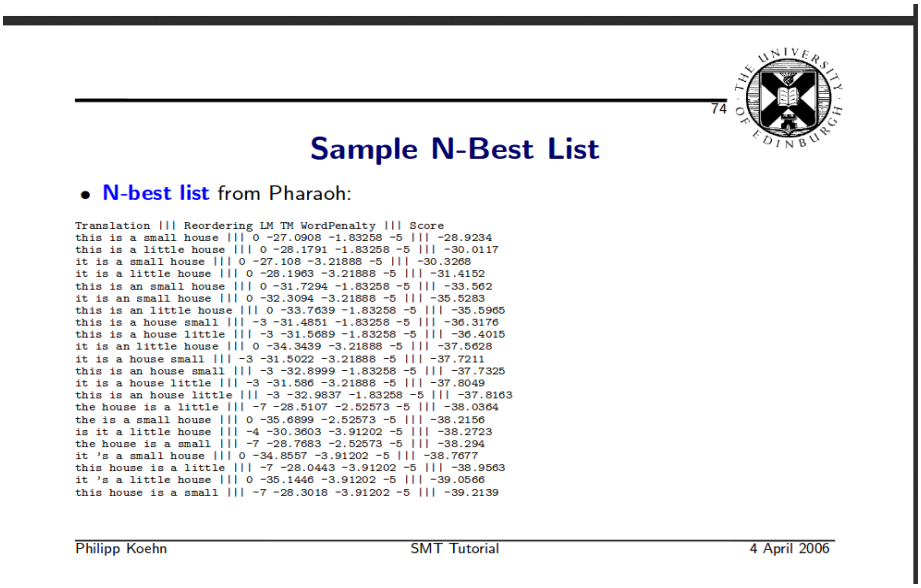


Figure 10 — Example of an n-best list

It's like making a puzzle. Some pieces are easier to fit than others!

In SMT, sometimes Moses translates segments amazingly well — even difficult ones -... and sometimes its translation is plain rubbish!

It all depends on a large number of factors, an important one being the relevance of the data the system has been trained with to the document(s) to be translated.

A large amount of data is important, but equally important is to train the system with relevant (in-domain) data.

In MMM, you can select some settings that may, or not, improve the quality of MT output for a particular language pair and type of documents to translate.

1.11. Automatic scoring of MT output

The **golden rule** for MT output evaluation is **human evaluation** and there is lots of literature on that on the Internet.

But there is nothing better than training a corpus and see how useful it is — in real terms — for your translation work!

However, as for research and production purposes, human evaluation is time-consuming and expensive, various metrics (tests) have been developed to automatically score MT output.

The BLEU score is one of the most widely used and it indicates how closely the word sequences in one set of data correlate with (match) the word sequences in another set of data, such as a reference human translation. The higher the score the better.

This means that if MT output is identical or near the human translation, the score will reflect it. However, it doesn't always work the other way round as there are generally many different ways of translating a sentence and, just because the MT translation is (very) different from the reference human translation, it doesn't necessarily mean that it is worse or incorrect!

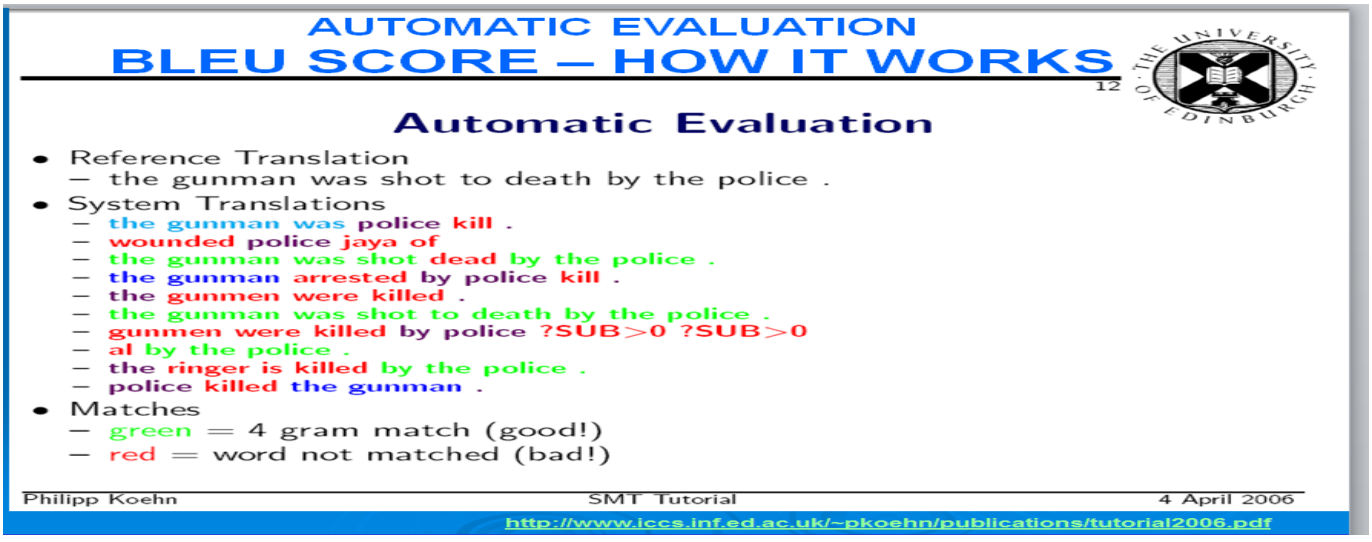


Figure 11 — The BLEU score — How it works

In MMM, the results are presented on a scale of 0 to 1 as generated by the scorer, but it is easier to think of it in a scale of 1 to 100. Therefore, a 0.5685 BLEU score may be read as a kind of percentage: 56.85.

In MMM, it is also available the NIST score.

To have an idea of the quality of MT output — as measured by the BLEU metrics, see the paper 462 Machine Translation Systems for Europe, by Philipp Koehn, Alexandra Birch and Ralf Steinberger (2009)[25].

SMT has evolved since then, but this gives an idea of how different the MT quality levels can be depending — among other factors — on the language combination.

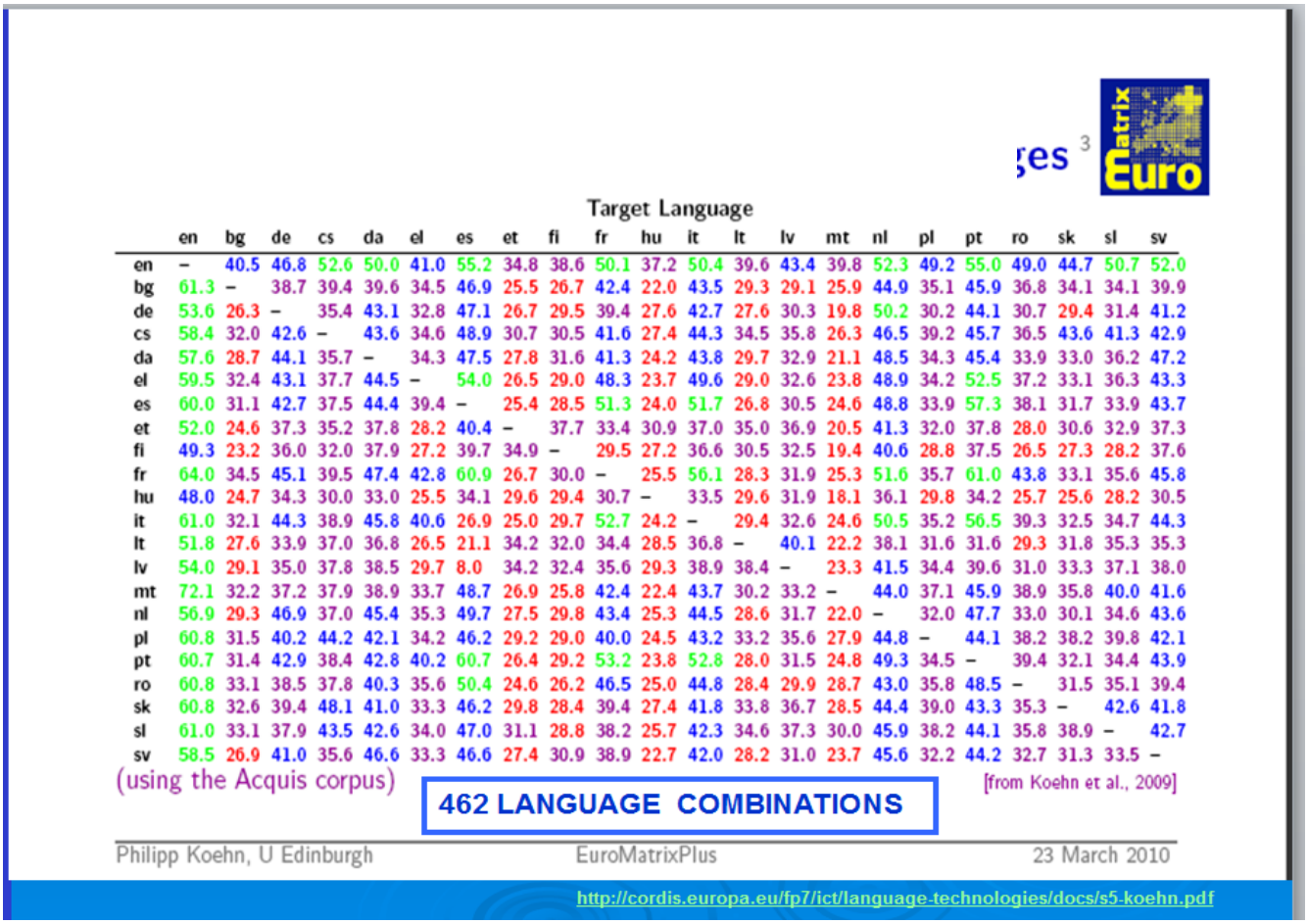


Figure 12 — Translating between EU official languages

[25] http://www.mt-archive.info/MTS-2009-Koehn-1.pdf

ANNEX 2 — List of MMM default settings

1. *Install* script

mosesdir="$HOME/Desktop/Machine-Translation/MMM"

2. *Create* script

mosesdir="$HOME/Desktop/Machine-Translation/MMM"
MMMdir="$HOME/Desktop/Machine-Translation/Moses-for-Mere-Mortals"
corpora_trained="$mosesdir/corpora_trained"
cores=
use_local_packages=1
report=
install_irstlm=1
install_randlm=1
install_mgiza=1
install_moses=1
install_scripts=1
install_scorers=1
dont_check_size=0

3. *make-test-files* cript

mosesdir="$HOME/Desktop/Machine-Translation/MMM"
lang1=pt
lang2=en
corpusbasename=200000
tuning_totalnumsectors=100
tuning_numsegs=10
test_totalnumsectors=100
test_numsegs=10

4. Train script

mosesdir="$HOME/Desktop/Machine-Translation/MMM"

4.1 Languages

lang1=pt
lang2=en

4.2 Files

corpusbasename=200000.for_train
lmbasename=300000
recaserbasename=300000
tuningbasename=200000.for_tuning
testbasename=200000.for_test

4.3 Training steps

cores=
reuse=1
paralleltraining=1
firsttrainingstep=1
lasttrainingstep=9
memmapping=1
runtrainingtest=1

4.4. Language model parameters

lngmdl=1
Gram=7

4.4.1. IRSTLM parameters

distributed=1
dictnumparts=20
s='improved-kneser-ney'
quantize=0
lmmemmapping=1

4.4.2. RandLM parameters

inputtype=corpus
falsepos=8
values=8

4.5. Training steps

4.5.1. Training step 1

nummkclsiterations=2
numclasses=50

4.5.2. Training step 2

4.5.2.1. GIZA parameters

ml=101
model1iterations=5
model2iterations=0
hmmiterations=5
model3iterations=3
model4iterations=3
model5iterations=0
model6iterations=0
countincreasecutoff=1e-06
countincreasecutoffal=1e-05
mincountincrease=1e-07
peggedcutoff=0.03
probcutoff=1e-07
probsmooth=1e-07
compactalignmentformat=0
model1dumpfrequency=0
model2dumpfrequency=0
hmmdumpfrequency=0
transferdumpfrequency=0
model345dumpfrequency=0

nbestalignments=0
nodumps=1
onlyaldumps=1
verbose=0
verbosesentence=-10
emalsmooth=0.2
model23smoothfactor=0
model4smoothfactor=0.4
model5smoothfactor=0.1
nsmooth=4
nsmoothgeneral=0
compactadtable=1
deficientdistortionforemptyword=0
depm4=76
depm5=68
emalignmentdependencies=2
emprobforempty=0.4
m5p0=-1
manlexfactor1=0
manlexfactor2=0
manlexmaxmultiplicity=20
maxfertility=10
p0=0.999
pegging=0

4.5.3. Training script parameters

alignment=grow-diag-final-and
reordering=msd-bidirectional-fe
MinLen=1
MaxLen=60
MaxPhraseLength=7

4.5.4. Decoder parameters

weight_l=1
weight_d=1
weight_w=0
mbr=0
mbrsize=200
mbrscale=1.0
monotoneatpunctuation=0
ttablelimit=20
beamthreshold=0
earlydiscardingthreshold=0
searchalgorithm=1
cubepruningpoplimit=2000
stack=2000
maxphraselen=20
cubepruningdiversity=0
distortionlimit=6

4.6. Tuning parameters

tuning=no-tuning
maxruns=25
tuning_type=mert
previous_weight_file="weight.ini"

5. Translate script

mosesdir="$HOME/Desktop/Machine-Translation/MMM"
report_file=
create_translation_report=1
cores=
alignment=grow-diag-final-and
reordering=msd-bidirectional-fe
weight_l=1
weight_d=1
weight_w=0

mbr=0
mbrsize=200
mbrscale=1.0
monotoneatpunctuation=0
ttablelimit=20
beamthreshold=0
earlydiscardingthreshold=0
searchalgorithm=1
cubepruningpoplimit=2000
stack=2000
maxphraselen=20
cubepruningdiversity=0
distortionlimit=6

6. *Score* script

mosesdir="$HOME/Desktop/Machine-Translation/MMM"
cores=
batch_user_note="2014"

7. *transfer-training-to-another- location* script

mosesdirmine="$HOME/Desktop/Machine-Translation/MMM"
newusername=john
mosesdirotheruser="$HOME/Desktop/Machine-Translation/moses-irstlm-randlm"

Detailed Table of Contents