

iSenWeb: A Web-based Machine Translation System to Translate Sentences

Introduction

The rapid development of Internet has made many changes in the daily life. As more and more digitized multilingual information are available in the Internet, increasingly people not only want to access information different from their mother language, but also want to have a translation platform for doing the translation. Indeed, World Wide Web provides an ideal and great environment for Machine Translation (MT) [1].

Today web-based machine translation is widely used for instantly translating words, phrases, sentences, documents, and websites. The social role of the web-based machine translation has gained much importance in people's global interaction because it allows them to communicate with anyone, from anywhere, in any language.

Moses is a cutting-edge machine translation program that reflects the latest developments in the area of statistical machine translation research. It can be trained to translate between any two languages, and yields high quality results. There is already a program written by *Herve Saint-Amand* to deal with the websites level translation in Moses document [2]. In this document, I will describe a program that can be used to translate the given plain text on sentence level one by one like Google translator [3] or the demo part in Moses [4].

System Description

Machine Translation (MT) is a field of computational linguistics that investigates the translation of texts from one human language to another, while SMT, in contrast to many automatic rule-based translation systems, is a translation paradigm based on statistical learning techniques using statistical information drawn from the large parallel training corpus. In most cases, the training corpus and training results are too large to be adapted for personal desktop computers. However, the web-based MT can overcome this disadvantage by the client server pattern. The direct goal of this system is to return the translation results with the help of existing tools, such as Moses toolkit. This system is basically an extension of the web translation which used to translate the webpage given the URL as described in Moses document [2]. To bring in the Moses toolkit, one script (*moses.pl*) will be used to activate the Moses server, which

is the same as introduced by *Herve Saint-Amand* .

The system here is to reach two goals:

- i. Enabling users or researchers to create their own web-based machine translation using existing Moses tools.
- ii. Improve the response time (or the time of returning the translation result) with the help of PHP and Ajax. The CGI method will not be used in this system.

System Architecture

The architecture of the system is simple (presented in Figure 1): a client requests a translation via the apache server, and then the translation will be dealt with in the Moses server. Different language pairs can distributed in different Moses servers. For example, Moses server 1 only deals with English-Chinese translation tasks and Moses server 2 can only translate the English-French language pairs.

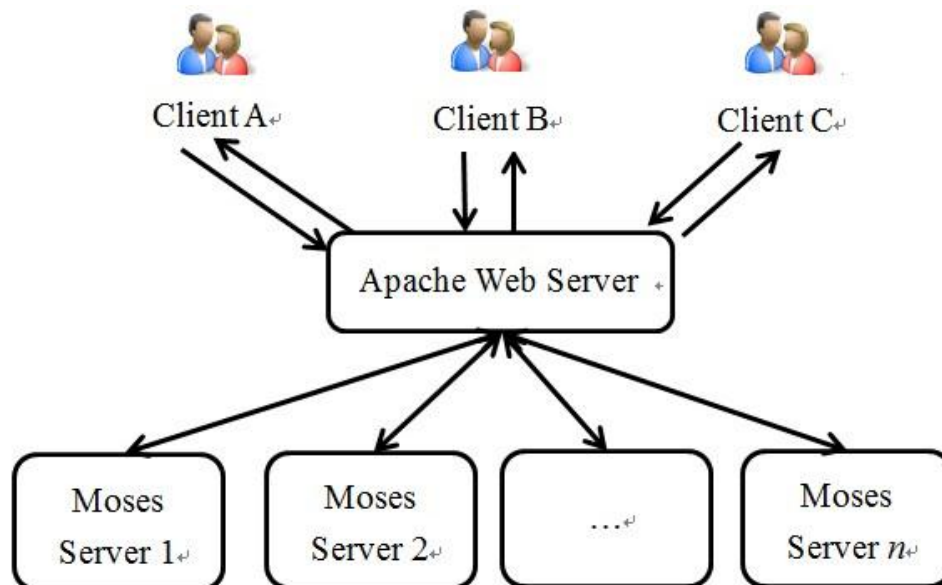


Figure 1. Web Based MT Architecture Based on Multiple Moses Servers

There are five files or folders in the package, which shown in Figure 2. The following is a quick overview of how the whole system works.

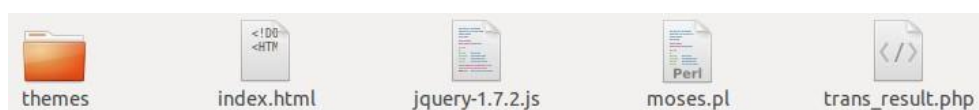


Figure 2. Files in the iSenWeb Package

- (1) The Moses system is installed and configured on one or several computers. Different Moses can focus on different language pairs translation task.
- (2) On each Moses server, a script, implemented by *moses.pl*, accepts network connections on a given port and copies everything it gets from those connections

straight to Moses (Obtain the input sentences through PHP programming), sending back to the client what Moses printed back (Print the translation result using Ajax in the user interface).

- (3) Another computer, which designates as the **web server**, runs Apache (or similar) Web server software.
- (4) Through that server, the PHP script *index.html* will be served to the client, providing the user interface to the system. In the interface, clients can requests *translating texts* via the Web server. And then the translated text will be returned after the translation by Moses by the file of *trans_result.php*. The result will be presented in the same page by the Ajax technique.

Note: The *index.html* will requires the script of *jquery-1.7.2.js*.

Detailed setup instructions

- I. **Testing Environment:** Ubuntu Server 12.04 LTS
- II. **Programming Language:** PHP and perl
- III. **Software Installation**

(1) Moses Installation

Simple Introduction:

Download: *git clone git://github.com/moses-smt/mosesdecoder.git*

Install: The install process is much easier in the newest version. The process is like below:

```
bjam --with-srilm=/path/to/the/srilm --with-irstlm=/path/to/the/irstlm  
--with-xmlrpc-c=/path/to/the/xmlrpc-c
```

Note: the giza-bin includes, *GIZA++*, *mkcls*, *snt2cooc.out*

Take my command as an example:

```
/bjam--with-srilm=/home/tianliang/research/srilm  
--with-irstlm=/home/tianliang/research/irstlm  
--with-xmlrpc-c=/home/tianliang/research/stable
```

(2) Apache and PHP Installation

We can simply install the apache in Ubuntu in command:

```
sudo apt-get install apache2
```

Of course if you want the full components which can be fit for PHP, you should type in command:

```
sudo apt-get install apache2 libapache2-mod-php5
```

In my program, I use the second way, because some of the scripts are written in PHP.

(3) Netcat Installation

The toolkit is used to listen on the given IP and port. This tool is very important, if you do not have this one installed, the system can not return the translation result.

Simply type: *sudo apt-get install netcat*

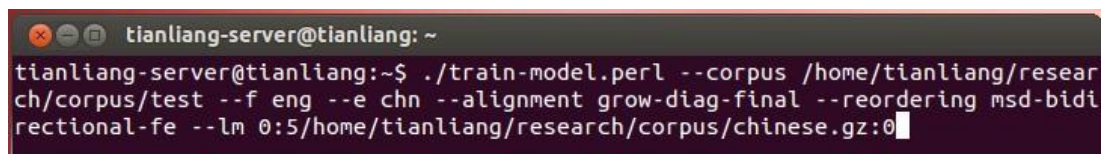
How to use the source code:

(1) Suppose you have successfully installed the Moses. There will be some executable files generated in the directory named *dist* in the newest version of Moses: *~dist/bin/moses* (or *~dist/bin/mosesserver*). In my lab's ([NLP²CT](#)) system I adapt the *moses* not the *mosesserver*.

Here I will take the *English-Chinese* language pair as an example.

Firstly, go to the folder *~/scripts/training/* and train the given language pair in command:

```
./train-model.perl --corpus /home/tianliang/research/corpus/test --f eng --e chn  
--alignment grow-diag-final --reordering msd-bidirectional-fe --lm  
0:5/home/tianliang/research/corpus/chinese.gz:0
```



A terminal window titled 'tianliang-server@tianliang: ~' showing the command `./train-model.perl --corpus /home/tianliang/research/corpus/test --f eng --e chn --alignment grow-diag-final --reordering msd-bidirectional-fe --lm 0:5/home/tianliang/research/corpus/chinese.gz:0` being executed.

Secondly, copy the *moses* from *~/bin/moses* to *~/scripts/training/model*

(2) Open the *moses.pl* in the package and then edit the scripts to point out where your Moses (*moses*) and configure file (*moses.ini*) are located.

```
my $MOSES      = '/home/tianliang/research/moses-smt/scripts/training/model/moses';
```

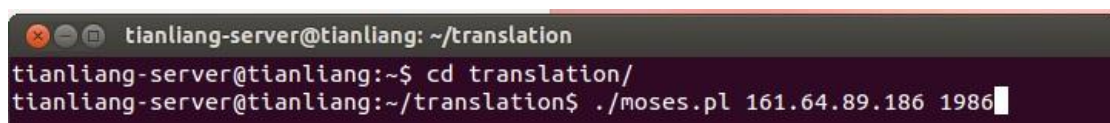
```
my $MOSES_INI  = '/home/tianliang/research/moses-smt/scripts/training/model/moses.ini';
```

To activate a Moses server, simply type, in a shell running on that server

```
./moses.pl <hostname> <port>
```

where *<hostname>* is the name of the host you're typing this on (found by issuing the *hostname* command), and *<port>* is the port you selected. In our lab server, I use the hostname **161.64.89.186** and port **1986**, which can be like below:

```
./moses.pl 161.64.89.186 1986
```



A terminal window titled 'tianliang-server@tianliang: ~/translation' showing the command `./moses.pl 161.64.89.186 1986` being executed.

After that the Moses server will be activated and waits for the input sentences to translate.

```
Finished loading phrase tables : [14.000] seconds
IO from STDOUT/STDIN
Created input-output object : [14.000] seconds
```

Before opening the webpage, you can test whether the Moses can work. If you have the NetCat tool installed, you can test whether it worked by going to a shell on the Web server and typing `echo "I am a man" | nc <hostname> <port>`, and `<hostname>` and `<port>` by the values pointing to the Moses server you just set up. A translation should be printed back. An example in my server is:

echo " I am a man" | nc 161.64.89.186 1986

Now it will return the translation: 我是一个男人

(3) Now we can transfer the source code package into the server, which should be located in `/var/www/` as default in Ubuntu. And then, you can open a webpage by the URL:

<http://161.64.89.186/translation/index.html>



Figure 2. The Screenshot of the iSenWeb System

OK, test your system according to the steps described above!

Note:

- (1) The system can not translate the given text which contain the Apostrophe (') Quotation marks (\" and \"'\") and bracket (\"(\"). Of course you can deal with the input sentence firstly. Here I just show you a method how to build a web-based SMT system with the existing scripts and systems. You can perfect the system and beautify the system according to your own purpose.

- (2) If you want to let the MoSES Server run in your server all the time, you can type the command ends with a "&" : ./moses.pl 161.64.89.186 1986 &
- (3) The system is developed with the help of Jeff (JavaScript) and Vincent (provides the CSS).

References:

- [1] C.C. Hao, K.K. Leung, H.W. Tou, F. Wong, and Y.P. Li, "Knowledge Sharing in Network Based Portuguese-Chinese Translation System", Proceedings of Symposium on Applied Science and Technology in Macau, Macau, China, 2004, pp. 43–51.
- [2] Web-based Translation: <http://www.statmt.org/moses/?n=Moses.WebTranslation>
- [3] Francisco O., F. Wong, S. Chao, Pui-Chi Fong, 2011. Proceedings of 2011 International Conference on System Science and Engineering, Macau, China - June 2011
- [4] Moses demo: <http://www.statmt.org/moses/?n=Public.Demos>