Model Building from Bitext

1 Introduction

Modern SMT systems typically depend on three kinds of models: a language model, a translation model (aka phrase table), and lexicalized reodering model (aka lexicalized distortion model). The latter two require a parallel corpus (aka bitext) to be built. Building those bitext models requires a number of steps:

1. Word Alignment: Here word-to-word (but not necessarily one-to-one) links are learned for each parallel sentence pair in the the bitext.
2. Word Translation Models: Here models of the form $p(w_f|w_e)$ and $p(w_e|w_f)$ are learned, where w_e and w_f are words.
3. Lexicalized Reordering Models: These are models of the form $p(Orientation|(p_f, p_e))$ where Orientation=monotonic,swap,discontinuous, where p_f and p_e are phrases.
4. Translation Models: These are models of the form $p(p_f|p_e)$ and $p(p_e|p_f)$, where p_f and p_e are phrases.

With our model building tool you can do step 2 to 4 assuming that you already have a word aligned parallel corpus in hand. All these steps can be done by a single script.

`script/training/modelBuilding/build-models-from-wordAligned-bitext.pl`

a sample calling of the script is as follows:

```
./build-models-from-wordAligned-bitext.pl --input-files-prefix=aligned
--experiment-dir=./expdir  --dependencies=./dependencies --f=chinese --e=english
--a=grow-diag-final --build-distortion-model --use-dlr
--moses-orientation  --build-phrase-table >& err.log
```

- `--input-files-prefix`

  Specifies the shared prefix of the names of source, target and alignment files. This means that these three input files should have shared prefix in their names. For examples

  `aligned.chinese, aligned.english and aligned.grow-diag-final.`

- `--experiment-dir`

  Specifies the path in which the input files are placed and the final and intermediate output files of the model building process will be created.

- `--f`

  Specifies the suffix of the input source file. For example,

  `--f=chinese`

  if the source file name is like

  `[FILENAME].chinese`

- `--e`

  Specifies the suffix of the input target file. For example,

  `--e=english`

  if the target file name is like

  `[FILENAME].english`

- `--a`

  Specifies the suffix of the input alignment file. For example,

  `--a=grow-diag-final`

  if the alignment file name is like

  `[FILENAME].grow-diag-final`

- `--build-distortion-model`

  Flag to build lexicalized reordering model.

- `--use-dlr`

  Flag to generate 4 reordering orientations instead of 3 by splitting discontinuous orientation into discontinuous left and discontinuous right.

- `--build-phrase-table`

  Flag to create phrase table (translation model).

- `--moses-orientation`

  Flag to use moses style orientations to build lexicalized reordering models. The default is to use Oister style which achieves higher BLEU score comparing to moses style.

- `--dependencies`

  Specifies the path to the dependencies folder where other scripts are located.

This will take some time, depending on the size of the bitext files. If they are small ($< 2,000$ lines, for debuggin purposes), it'll take a few minutes. If they are large ($> 200,000$ lines) it can take several hours. After it has finished, you should see a number of new directories under the path to

`--experiment-dir`

. The most important one is models/model which contains the following files:

- `dm_fe_0.75.gz`

  The lexicalized reordering model.

- `lex.e2f and lex.f2e`

  The word translation models.

- `phrase-table.gz`

  The translation model.

2 Installation

There is no need to install the model building tool itself. It works right after cloning from the repository. However, the dependencies should be addressed beforehand.

2.1 Dependecies

This model building tool has dependencies to three binary files from Moses translation system. These files include:

- consolidate
- extract
- score

In order to meet the dependencies, first install Moses translation tool following the installation instructions.

After installation, copy the aforementioned binary files from Moses installation directory to the following path:

`[PATH-TO-MODEL-BUILDING-TOOL-HOME]/dependencies/moses/bin/`

Now the tool can be used by calling it using the command given in the example above.