PHARAOH

Training

Philipp Koehn
koehn@csail.mit.edu
MIT CSAIL

April 14, 2004

Abstract

Handout to illustrate the training process of a phrase-based machine translation system. The training process is still a ship that is being build, but let's take it out to sea!

1 Training

The training process takes place in 7 steps, all of them executed by a script currently called
`/projects/nlp/koehn/bin/phrase-extract-MIT-v2.perl`

The seven steps are

1. prepare data (1:30 hours)
2. run GIZA++ (20 hours)
3. align words (3 hours)
4. get lexical translation table (30 min)
5. extract phrases (7 hours)
6. digest phrases (30 minutes)
7. create configuration file

The run times mentioned in the steps refer to a recent training run on the 751'000 sentence, 15 million word German-English Europarl corpus, on a 1GHz Linux machine. I am currently reimplementing step 5 and 6 in C++ (from the original Perl), which seems to give a speedup by factor 50.

Training data has to be provided sentence aligned (one sentence per line), in two files, one for the foreign sentences, one for the English sentences:

```
>head -3 corpus/clean.*
==> corpus/clean.de <==
wiederaufnahme der sitzungsperiode
ich erklaere die am donnerstag , den 28. maerz 1996 unterbrochene sitzungsperiode des europaeischen parlaments fuer wiederaufgenommen .
begruessung

==> corpus/clean.en <==
resumption of the session
i declare resumed the session of the european parliament adjourned on thursday , 28 march 1996 .
welcome
```

A few other points have to be taken care of:

- one sentence per line, no empty lines
- sentences longer than, say, 40 words (and their corresponding translations) have to be eliminated
- everything lowercased

The script may be executed as:

```
phrase-extract-MIT.perl --root-dir . --f de --e en --corpus corpus/clean --low-memory > & ! LOG &
```

The training data should be placed in a special directory called `corpus`. This directory and all other files are in the location specified by the `--root-dir` option. Here it is the current directory (`.`), but we should always specify a full path. The foreign and English corpus file names differ only in their extension, which is specified at the command line with the switches `--f` and `--e`.

1.1 Data Preparation

The parallel corpus has to be converted into a format that is suitable to the GIZA++ toolkit. Two vocabulary files are generated and the parallel corpus is converted into a numberized format.

The vocabulary files contain words, integer word identifiers and word count information:

```
==> corpus/de.vcb <==
1       UNK     0
2       ,       928579
3       .       723187
4       die     581109
5       der     491791
6       und     337166
7       in      230047
8       zu      176868
9       den     168228
10      ich     162745

==> corpus/en.vcb <==
1       UNK     0
2       the     1085527
3       .       714984
```

```
4       ,       659491
5       of      488315
6       to      481484
7       and     352900
8       in      330156
9       is      278405
10      that    262619
```

The sentence-aligned corpus now looks like this:

```
> head -9 corpus/en-de-int-train.snt
1
3469 5 2049
4107 5 2 1399
1
10 3214 4 116 2007 2 9 5254 1151 985 6447 2049 21 44 141 14 2580 3
14 2213 1866 2 1399 5 2 29 46 3256 18 1969 4 2363 1239 1111 3
1
7179
306
```

A sentence pair now consists of three lines: First the frequency of this sentence. In our training process this is always `1`. This number can be used for weighting different parts of the training corpus differently. The two lines below contain word ids of the foreign and the English sentence. In the sequence `4107 5 2 1399` we can recognize `of (5)` and `the (2)`.

GIZA++ also requires words placed into word classes. This is done automatically by calling the `mkcls` program. Word classes are only used for the IBM reordering model in GIZA++. A peek into the foreign word class file:

```
> head corpus/de.vcb.classes
!       14
"       14
#       30
%       31
&       10
'       14
(       10
)       14
+       31
,       11
```

1.2 Running GIZA++

GIZA++ is a freely available implementation of the IBM Models. We need it as a initial step to establish word alignments. Our word alignments are taken from the intersection of bidirectional runs of GIZA++ plus some additional alignment points from the union of the two runs.

Running GIZA++ is the most time consuming step in the training process. It also requires a lot of memory (1-2 GB RAM is common for large parallel corpora).

GIZA++ learns the translation tables of IBM Model 4, but we are only interested in the word alignment file:

```
> zcat giza.de-en/de-en.A3.final.gz | head -9
# Sentence pair (1) source length 4 target length 3 alignment score : 0.00643931
wiederaufnahme der sitzungsperiode
NULL ({ }) resumption ({ 1 }) of ({ }) the ({ 2 }) session ({ 3 })
# Sentence pair (2) source length 17 target length 18 alignment score : 1.74092e-26
ich erklaere die am donnerstag , den 28. maerz 1996 unterbrochene sitzungsperiode
  des europaeischen parlaments fuer wiederaufgenommen .
NULL ({ 7 }) i ({ 1 }) declare ({ 2 }) resumed ({ }) the ({ 3 }) session ({ 12 })
  of ({ 13 }) the ({ }) european ({ 14 }) parliament ({ 15 })
  adjourned ({ 11 16 17 }) on ({ }) thursday ({ 4 5 }) , ({ 6 }) 28 ({ 8 })
  march ({ 9 }) 1996 ({ 10 }) . ({ 18 })
# Sentence pair (3) source length 1 target length 1 alignment score : 0.012128
begruessung
NULL ({ }) welcome ({ 1 })
```

In this file, after some statistical information and the foreign sentence, the English sentence is listed word by word, with references to aligned foreign words: The first word `resumption ({ 1 })` is aligned to the first German word `wiederaufnahme`. The second word `of ({ })` is unaligned. And so on.

Note that each English word may be aligned to multiple foreign words, but each foreign word may only be aligned to at most one English word. This one-to-many restriction is reversed in the inverse GIZA++ training run:

```
> zcat giza.en-de/en-de.A3.final.gz | head -9
# Sentence pair (1) source length 3 target length 4 alignment score : 0.000985823
resumption of the session
NULL ({ }) wiederaufnahme ({ 1 2 }) der ({ 3 }) sitzungsperiode ({ 4 })
# Sentence pair (2) source length 18 target length 17 alignment score : 6.04498e-19
i declare resumed the session of the european parliament adjourned on thursday , 28 march 1996 .
NULL ({ }) ich ({ 1 }) erklaere ({ 2 10 }) die ({ 4 }) am ({ 11 })
  donnerstag ({ 12 }) , ({ 13 }) den ({ }) 28. ({ 14 }) maerz ({ 15 })
  1996 ({ 16 }) unterbrochene ({ 3 }) sitzungsperiode ({ 5 }) des ({ 6 7 })
  europaeischen ({ 8 }) parlaments ({ 9 }) fuer ({ }) wiederaufgenommen ({ }) . ({ 17 })
# Sentence pair (3) source length 1 target length 1 alignment score : 0.706027
welcome
NULL ({ }) begruessung ({ 1 })
```

1.3 Align Words

To establish word alignments based on the two GIZA++ alignments, a number of heuristics may be applied. The default heuristic `grow-diag-final` starts with the intersection of the two alignments (here, for sentence 2):

```
#----------------- i
-#---------------- declare
------------------ resumed
--#--------------- the
-----------#------ session
```

```
------------#----- of
------------------ the
-------------#---- european
--------------#--- parliament
------------------ adjourned
------------------ on
----#------------- thursday
-----#------------ ,
-------#---------- 28
--------#--------- march
---------#-------- 1996
-----------------# .
iedad,d2m1usdepfw.
crimo e8a9nieuaui
hke n n.r9ttsrrre
 l  n   z6ez ol d
 a  e     ru pa e
 r  r     bn am r
 e  s     rg ie a
    t     os sn u
```

and adds some additional alignment points that lie in the union of the two alignments:

```
#----------------- i
-#---------------- declare
----------#------- resumed
--#--------------- the
-----------#------ session
------------#----- of
------------#----- the
-------------#---- european
--------------#--- parliament
---------------##- adjourned
---#-------------- on
---##------------- thursday
-----#------------ ,
-------#---------- 28
--------#--------- march
---------#-------- 1996
-----------------# .
iedad,d2m1usdepfw.
crimo e8a9nieuaui
hke n n.r9ttsrrre
 l  n   z6ez ol d
 a  e     ru pa e
 r  r     bn am r
 e  s     rg ie a
    t     os sn u
```

This alignment has a blatant error: the alignment of the two verbs is mixed up. `resumed` is aligned to `unterbrochene`, and `adjourned` is aligned to `wiederaufgenommen`, but it should be the other way around.

To conclude this section, a quick look into the files generated by the word alignment process:

```
==> model/aligned.de <==
wiederaufnahme der sitzungsperiode
ich erklaere die am donnerstag , den 28. maerz 1996 unterbrochene sitzungsperiode
  des europaeischen parlaments fuer wiederaufgenommen .
begruessung

==> model/aligned.en <==
resumption of the session
i declare resumed the session of the european parliament adjourned on
  thursday , 28 march 1996 .
welcome

==> model/aligned.grow-diag-final <==
0-0 0-1 1-2 2-3
0-0 1-1 2-3 3-10 3-11 4-11 5-12 7-13 8-14 9-15 10-2 11-4 12-5 12-6 13-7 14-8 15-9 16-9 17-16
0-0
```

The third file contains alignment information, one alignment point at a time, in form of the position of the foreign and English word.

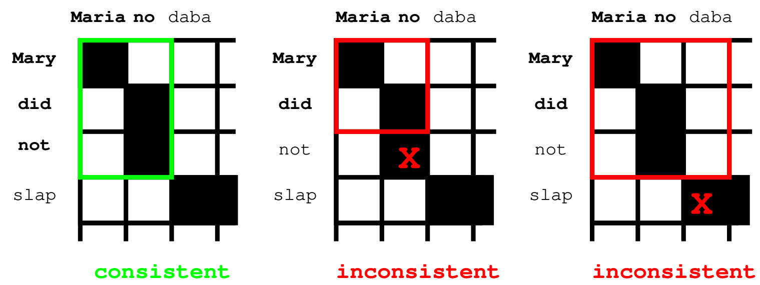
1.4 Get Lexical Translation Table

Given this alignment, it is quite straight-forward to estimate a maximum likelihood lexical translation table. We estimate the $p(e|f)$ as well as the inverse $p(f|e)$ translation table. Here are the top translations for `europa` into English:

```
> grep ' europa ' model/lex.f2n | sort -nrk 3 | head
europe europa 0.8874152
european europa 0.0542998
union europa 0.0047325
it europa 0.0039230
we europa 0.0021795
eu europa 0.0019304
europeans europa 0.0016190
euro-mediterranean europa 0.0011209
europa europa 0.0010586
continent europa 0.0008718
```

1.5 Extract Phrases

All phrase pairs that are consistent with the word alignment are collected. The definition of *consistent* can be best explained with an example:



A phrase alignment here is a box, and a word is either a row (English word) or a column (foreign word). For a phrase alignment to be consistent with the word alignment, all alignment points for rows and columns that are touched by the box have to in the box, and not outside.

In the second example the alignment point `no-not` is outside the box, although the word `no` is touched by the box. Hence, the box is an inconsistent word alignment.

In the phrase translation step, all phrases are dumped into one big file. Here is the top of that file:

```
> head model/extract
wiederaufnahme ||| resumption of ||| 0.38840215
wiederaufnahme der ||| resumption of the ||| 0.102352783253805
der ||| the ||| 0.2635227
der sitzungsperiode ||| the session ||| 0.11074957833366
sitzungsperiode ||| session ||| 0.4202658
ich ||| i ||| 0.7750742
ich erklre ||| i declare ||| 0.32787142303948
erklre ||| declare ||| 0.4230194
unterbrochene ||| resumed ||| 0.092296
die ||| the ||| 0.3130043
```

The content of this file is currently foreign phrase, English phrase, and lexical score. The lexical score will be soon replaced by the alignment points. As I said earlier, this is a bit fluid at this time.

1.6 Digest Phrases

In a second step, a translation table is created from the stored phrase translation pairs. The two steps are separated, because for larger translation models, the phrase translation table does not fit into memory. Fortunately, we never have to store the phrase translation table into memory — we can construct it on disk.

To estimate $p(e|f)$ we proceed as follows: First, the extract file is sorted. This ensures that all English phrase translations for an foreign phrase are next to each other in the file. Thus, we can process the file, one foreign phrase at a time, collect counts and compute $p(e|f)$ for that foreign phrase f.

To estimate $p(f|e)$, the file is inverted, sorted, and then $p(f|e)$ is estimated for an English phrase at a time.

Next to phrase translation probability distributions $p(f|e)$ and $p(e|f)$, additional phrase translation scoring functions can be computed, e.g. lexical weighting, word penalty, phrase penalty, etc. I am currently experimenting here, so no final word on this yet.

Here the phrase translation table for `in europa`:

```
> grep '| in europa |' model/phrase-table.f2n | sort -nrk 7 -t\| | head
in europe ||| in europa ||| 0.81004413 0.21706101
of europe ||| in europa ||| 0.04916999 0.01172112
within europe ||| in europa ||| 0.01702038 0.15586666
european ||| in europa ||| 0.01575961 0.00034347
in europe , ||| in europa ||| 0.01449884 0.21706101
europe ||| in europa ||| 0.01407859 0.01513660
across europe ||| in europa ||| 0.00840513 0.12115531
for europe ||| in europa ||| 0.00567346 0.01125292
throughout europe ||| in europa ||| 0.00546333 0.15832276
to europe ||| in europa ||| 0.00462282 0.01242431
```

The first number is the phrase translation probability $p(e|f)$, the second lexical weighting.

1.7 Create Configuration File

As a final step, a configuration file for the decoder is generated with all the correct paths for the generated model and a number of default parameter settings.