MLSALT 8: Statistical Machine Translation
Practical 3: Hierarchical Phrase-based Translation with alternative grammars

FEYNMAN LIANG

1. Preliminary Questions

1. The language model probability is not included as a feature in the rulefile because it is defined for N-grams over the target language (i.e. English). This means that they can only be applied when:
   (a) A complete sequence of terminals has been derived and no non-terminals are remaining
   (b) The identity of the previous $N-1$ words is known

   Rules containing non-terminals or yielding less than N terminal symbols do not satisfy these requirements, hence it doesn't make sense to assign them a language model score.

   The language model scores could be included if:
   (a) The language model is a 1-gram model, in which case the language model score of a rule is the joint probability of all the terminals derived one step after applying the rule
   (b) The degenerate case where the entire sequence of non-terminals is derived in a single rule. The score would then be the score of the target sentence under the language model.
2.
3.

```
printstrings -n 1000 -u -w --input=output/example/LATS.hyp1/14.fst.gz \
  --print-output-labels 2> /dev/null
```

2. First part

1. We translate the 30 sentences with grammar A:

```
hifst $DIR/configs/basic+params.features \
  --textinput=$DIR/input/test30.spa.idx \
  --rulefile=$GRAMA/r.?.gz \
  --lm=$DIR/lm/test30.news-newscomm.eng.4g/G/?.G.gz --lmn=4 \
  --range=1:30 \
  --latoutputfst=output/example/LATS.A/?.fst.gz

printstrings --r=1:30 --input=output/example/LATS.A/?.fst.gz \
  --output=outA --label-map=$SUNMAP
```

and grammar B:

```
hifst $DIR/configs/basic+params.features \
  --textinput=$DIR/input/test30.spa.idx \
  --rulefile=$GRAMB/r.?.gz \
  --lm=$DIR/lm/test30.news-newscomm.eng.4g/G/?.G.gz --lmn=4 \
  --range=1:30 \
  --latoutputfst=output/example/LATS.B/?.fst.gz

printstrings --r=1:30 --input=output/example/LATS.B/?.fst.gz \
  --output=outB --label-map=$SUNMAP
```

During this process, we found that the run with grammar B was significantly slower than the run with grammar A.

Computing BLEU scores:

```
print "Scoring grammar A:"
```

Date: March 16, 2016.

```
ScoreBLEU.sh \
  -t outA \
  -r $DIR/reference/test30.eng
 BLEU score = 0.3515 (0.3515 * 1.0000) for system "1"
 faster

print "Scoring grammar B:"
ScoreBLEU.sh \
  -t outB \
  -r $DIR/reference/test30.eng
```

We found that Grammar A attains a BLEU score of 0.3515 while grammar B achieves 0.3861.

2. (a)
   (b)
3. Some main differences include:
   (a) A has 104 rules B has 321
   (b) A's rules only contain word and phrasal translations; hiero rules (i.e. productions with both terminals and non-terminals in the yield) are absent. This means that A cannot model arbitrarily long context (i.e. has as distortion limit) and is equivalent in expressivity to a phrase-based SMT system.
   In contrast, B contains hiero rules out of the X non-terminal and hence implement hierarchical phrases, which have greater generality but also increased computational complexity.

   These differences can help explain differences in translation. For reference, sentence 27:

   ```
   y después llegó la época americana .
   ```

   is translated uder grammar A to:

   ```
   <s> and then came the time american . </s>
   ```

   while under grammar B is translated to:

   ```
   <s> and then came the american era . </s>
   ```

   The grammar A translation appears to translate the sentence word-for-word, translationg "epoca" literally to time because it failed to account for the context. In contrast, grammar B correctly accounts for the context, translating "epoca americana" to "american era."

   The reason why grammar B is able to account for context lies in the presence of non-terminals in its yields, ultimately allowing it to achieve a significantly higher BLEU score. On the other hand, it also explains why translating under grammar B takes more time than grammar A: the presence of non-terminals significantly increases the complexity of decoding because it leverages the full generality of Hierarchical Phrase Based Translation.
4. Figure 1 and Figure 2 show the derivation trees for sentences 27 under rulesets A and B respectively.

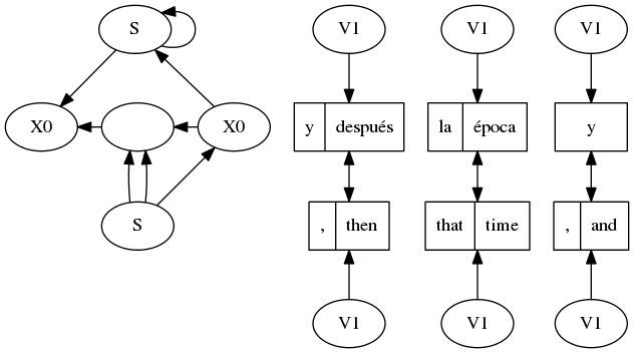

FIGURE 1. Sentence 27 derivation tree under ruleset A

5. Aligning the 30 sentences towards respective English references:

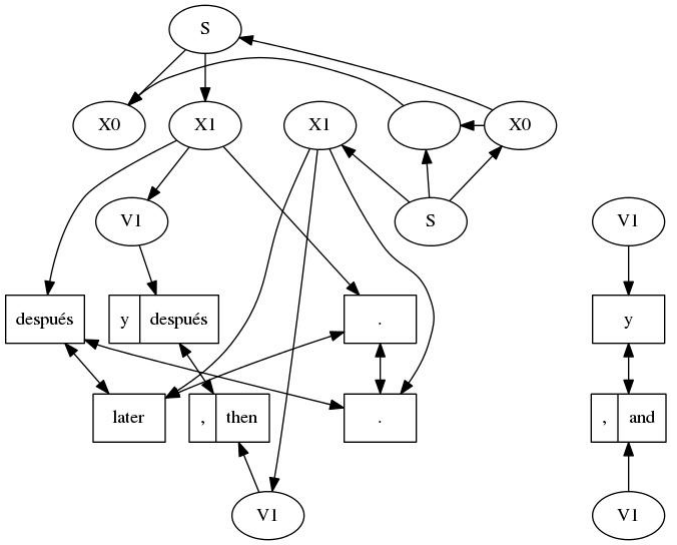

FIGURE 2. Sentence 27 derivation tree under ruleset B

```
hifst $DIR/configs/basic+params.features \
  -- textinput=$DIR/input/test30.spa.idx \
  -- rulefile =$GRAMA/r.?.gz \
  --lm=$DIR/lm/test30.news-newscomm.eng.4g/G/?.G.gz --lmn=4 \
  --range=1:30 \
  -- latoutputfst =output/example/LATS.A.towards_ref/?.fst.gz \
  -- towardsreference=$DIR/reference/test30/r.?.eng.idx
hifst $DIR/configs/basic+params.features \
  -- textinput=$DIR/input/test30.spa.idx \
  -- rulefile =$GRAMB/r.?.gz \
  --lm=$DIR/lm/test30.news-newscomm.eng.4g/G/?.G.gz --lmn=4 \
  --range=1:30 \
  -- latoutputfst =output/example/LATS.B.towards_ref/?.fst.gz \
  -- towardsreference=$DIR/reference/test30/r.?.eng.idx
```

Comparing the number of input sentences generating the reference for each grammar:

```
integer Acnt=0
integer Bcnt=0
for i in {1..30}; do
  integer newA=$(printstrings -n 500000 -u -w --input=output/example/LATS.A.towards_ref/$i.fst.gz \
    2>/dev/null \
    | wc -l)
  integer newB=$(printstrings -n 500000 -u -w --input=output/example/LATS.B.towards_ref/$i.fst.gz \
    2>/dev/null \
    | wc -l)
  print "$i, $newA, $newB"
  Acnt+=newA
  Bcnt+=newB
done
print "Acnt: $Acnt, Bcnt: $Bcnt"
```

We obtain the results shown in **??**.

6.
7.

3. SECOND PART

1.
2. Aligning the sentences with their English refrence with grammar C:

```
hifst $DIR/configs/basic+params.features \
  -- textinput=$DIR/input/test30.spa.idx \
```

Sentence #	Number inputs generating reference	
	Grammar A	Grammar B
1	4	8
2	1	1
3	1	1
4	1	1
5	1	165
6	1	1
7	1	8586
8	1	1
9	1	1
10	48	122
11	1	1
12	1	84
13	11070	51692
14	47	83
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	52	166
21	1	1
22	500000	500000
23	2586	14030
24	1	1
25	1	282
26	270	658
27	1	1
28	1	1
29	1	1
30	1	1
Total	514099	575894

TABLE 1. Sentences aligned towards their references

```
  -- rulefile =$GRAMC/r.?.gz \
  --lm=$DIR/lm/test30.news-newscomm.eng.4g/G/?.G.gz --lmn=4 \
  --range=1:30 \
  -- latoutputfst =output/example/LATS.C.towards_ref/?.fst.gz \
  -- towardsreference=$DIR/reference/test30/r.?.eng.idx
```

Comparing the number of input sentences generating the reference for grammars B and C:

```
integer Bcnt=0
integer Ccnt=0
for i in {1..30}; do
  integer newB=$(printstrings -n 500000 -u -w --input=output/example/LATS.B.towards_ref/$i.fst.gz \
    2>/dev/null \
    | wc -l)
  integer newC=$(printstrings -n 500000 -u -w --input=output/example/LATS.C.towards_ref/$i.fst.gz \
    2>/dev/null \
    | wc -l)
  print "$i, $newB, $newC"
  Bcnt+=newB
  Ccnt+=newC
done
print "Bcnt: $Bcnt, Ccnt: $Ccnt"
```

We obtain the results

3.