MPhil in Advanced Computer Science
Module L102: Statistical Machine Translation

Practical Handout 3
Hierarchical Phrase-based Translation with alternative grammars

1 Introduction

This third practical investigates the process of decoding with a hierarchical phrase-based statistical translation system with various translation models. For this purpose, the `HiFst` decoder and a set of relevant models are provided. A set of 30 sentences in Spanish is to be translated into English using 2 different translation grammars and parameters. The relevant files are in:

```
DIR=/usr/groups/acs-software/L102/practical-3

$DIR/input/test30.spa
$DIR/input/test30.spa.idx
$DIR/reference/test30.eng
$DIR/reference/test30.eng.idx
```

It is assumed that the tools used in the first two practicals are to be used here when necessary.
Important: Run the following command in order to setup the variables needed to run the practical.

```
source /usr/groups/acs-software/L102/setup.sh
```

1.1 Preliminary example

Before starting with the practical itself, a preliminary example is presented here in order to introduce the basic tools that will be used in this practical. Let us translate one single sentence from a given input text. A call to the `HiFst` translation system requires the following parameters:

- An **input file** to be translated (see example file in `$DIR/input/test30.spa.idx`). This contains the text to be translated, where each word has been mapped to an integer (see the text-format example file in `$DIR/input/test30.spa`). Each line in the input file is translated independently and will generate a different output lattice.
- A **range** value that determines which lines from the input file are to be translated. By default, only the first line is translated. For example, if we are interested in translating lines 6 and 7, we can define `--range=6:7`
- A **translation model**, expressed as a set of translation rules contained in a file (see example rule-file in `$DIR/rules/test30/B/r.14.gz`). Each line i in this file corresponds to a hierarchical phrase-based translation rule r_i of the format $N \rightarrow \langle source, target \rangle \; f_1(r_i) f_2(r_i) ... f_K(r_i)$,

where N is the non-terminal that generates the source and target sequences of words/nonterminals with feature scores f_1 to f_K according to the K submodels that comprise the translation model. For example, the following line in the rulefile:

```
X 296_V1_7 51_20_V1_5 -4.4 -3.0 3 1 0 0 0 0 0 1 -3.7 -9.5
```

is representing the rule $X \rightarrow \langle 296\ V\ 7, 51\ 20\ V\ 5 \rangle$, where words have been mapped to integers and X and V are nonterminals, and where we can see the scores of the 12 feature functions used by the translation model.

- A **parameter file**. This defines the scaling factor that needs to be applied to each feature function (see example file in `$DIR/config/params.features`). It has 12 scaling factors for the features found in the rulefile.
- A **language model**. This is a standard N-gram language model for the target language, in this case English. Words are also mapped to integers.
- An **output directory** where the translation lattices will be generated. Note that each line in the input sentence will generate an output lattice.

Preliminary Question 1: Examine the examples files mentioned above. Why is the language model probability not included as one of the features in the rulefile, together with the rest of the 12 feature functions? In which case could it be included?

For a preliminary test, run the following command, which translates sentence 14 from the test file:

```
hifst $DIR/configs/params.features \
--source.load=$DIR/input/test30.spa.idx \
--grammar.load=$DIR/rules/test30/B/r.?.gz \
--range=14:14 \
--lm.load=$DIR/lm/test30.news-newscomm.eng.4g/G/?/?.G.gz \
--hifst.lattice.store=output/example/LATS/?.fst.gz \
--hifst.prune=9
```

Note: The translation model files that will be used in this practical have been obtained from a large parallel corpus, which was automatically aligned at the word level, and from which rules were extracted according to standard heuristics. For simplicity, the corpus-level general rulefile (which is huge!) has been divided into several files, one for each input sentence. A similar process has been carried out for the language model. For this reason, the input rulefile and lms depend on the actual line being translated (i.e. the question mark `?` takes the value specified by the option `--range`).

Run the following command to check that the 1-best translation has a cost of 33.7 and is:

```
SUNMAP=$DIR/wmaps/english.unmap
zcat output/example/LATS/14.fst.gz | fstprintstrings -n 1 -u -c -1 -i $SUNMAP

<s> " these negotiations have been going on for two years . </s>  33.7925339
```

Preliminary Question 2: How many alternative translations are generated? How many translation candidates, including repeated hypotheses?

Apart from the 1-best translation hypothesis, we may be interested in which set of rules (or derivation) were used by the system to generate this particular hypothesis. In order to obtain the derivation, the `HiFst` decoder accepts an additional option:

- A **reference lattice** with one or more hypotheses that must be generated (option `--referencefilter.load`). If this option is active, then `HiFst` seeks to generate any of the specified reference hypotheses, and produces all sequences of translation rules that can generate these references. As usual, the English references must be integer-mapped.

For example, let us find the derivations that lead to the 1-best candidate from before.

```
hifst $DIR/configs/params.features \
--source.load=$DIR/input/test30.spa.idx \
--grammar.load=$DIR/rules/test30/B/r.?.gz \
--range=14:14 \
--lm.load=$DIR/lm/test30.news-newscomm.eng.4g/G/?/?.G.gz \
--hifst.lattice.store=output/example/LATS.hyp1/?.fst.gz \
--referencefilter.load=output/example/LATS/?.fst.gz \
--referencefilter.prunereferenceshortestpath=1 \
--hifst.alilatsmode
```

The output of the system is now a transducer, and we can print the input or output strings with usual FST commands:

```
zcat output/example/LATS.hyp1/14.fst.gz | fstprintstrings -n 1 -1 -u -c
385 3 1 42 1 70 1 99 1 20 1 8 1 93 1 185 2  33.7925339
```

returns the 1-best derivation (sequence of rules) that produces any of the references contained in `output/example/LATS/1.fst.gz`. Rules are mapped to numbers corresponding to the line number in the input rulefile where each rule is (see `$DIR/rules/test30/B/r.14.gz`).

```
zcat output/example/LATS.hyp1/14.fst.gz | fstprintstrings -n 1 -2 -u -c
1 215 57 380 23 47 310 17 14 141 137 5 2  33.7925339
```

returns the 1-best English translation found (in integer-mapped form), which should correspond to the 1-best of `output/example/LATS/14.fst.gz`.

Preliminary Question 3: Check that the hypothesis lattice has a single output translation. How many alternative derivations can generate this single hypothesis?

Finally, we can draw[1] any derivation in a tree-like structure as follows:

```
zcat output/example/LATS.hyp1/14.fst.gz | \
    fstprintstrings -n 1 -1 -u > example.dvn1

cat example.dvn1
385 3 1 42 1 70 1 99 1 20 1 8 1 93 1 185 2

draw_tree.sh example.dvn1 $DIR/rules/test30/B/r.14.gz tree14dvn1.jpg
```

[1] Note that `draw_tree.sh` accepts 'jpg, 'png', 'pdf' or 'ps' as output file extension.

The result is a graph as the one shown in Figure **??**. This shows the source and target trees obtained by the translation synchronous grammar when parsing the source sentence. Source and target terminals (words) appear within rectangles and are linked to each other as specified by the rules, whereas non-terminals appear within ovals. For visualization purposes, all non-terminals immediately below the S non-terminal are named X_y.

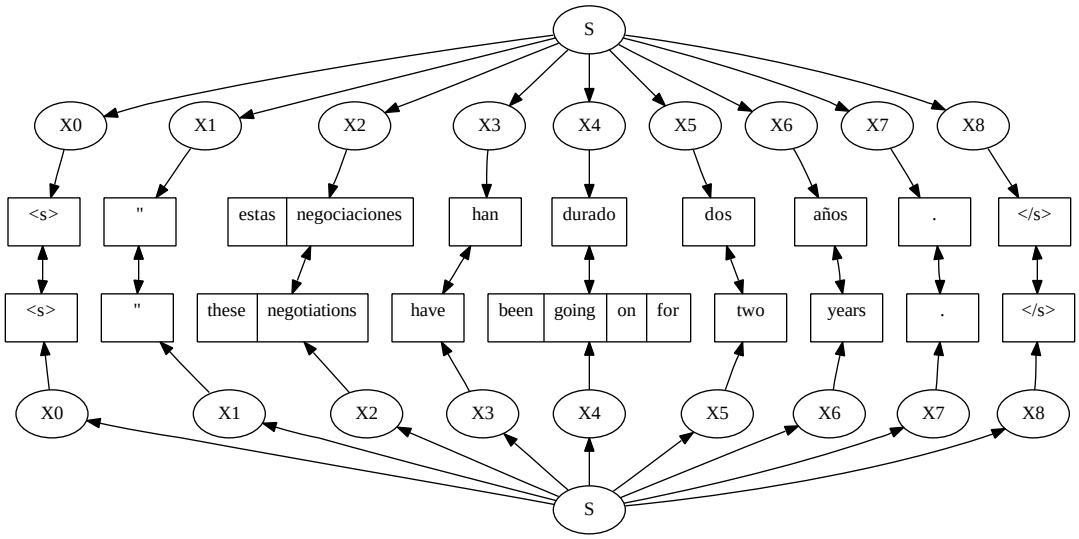

Figure 1: Source and target trees obtained for the first derivation in sentence 14 using grammar B.

The previous script also generates a file where the actual rules used in the tree can be seen:

```
cat tree14dvn1.jpg.rules

X <s> <s> 0 0 0 0 0 0 0 0 0 0 0 0
S S_X S_X
V " " 1.0 0.2 -1 -1 0 0 0 0 0 -1 1.0 0.1
X V V 0 0 0 0 0 0 0 0 0 0 0 0
S S_X S_X 0 0 0 0 1.0 0 0 0 0 0 0 0
V estas_negociaciones these_negotiations 0.3 0.7 -2.0 -1 0 0 0 0 0 -1 2.7 3.6
X V V 0 0 0 0 0 0 0 0 0 0 0 0
S S_X S_X 0 0 0 0 1.0 0 0 0 0 0 0 0
V han have 0.4 1.2 -1 -1 0 0 0 0 0 -1 0.6 1.8
X V V 0 0 0 0 0 0 0 0 0 0 0 0
S S_X S_X 0 0 0 0 1.0 0 0 0 0 0 0 0
V durado been_going_on_for 4.0 2.4 -4.0 -1 0 0 0 0 -1 0 9.4 17.5
X V V 0 0 0 0 0 0 0 0 0 0 0 0
S S_X S_X 0 0 0 0 1.0 0 0 0 0 0 0 0
V dos two 0.2 0.2 -1 -1 0 0 0 0 0 -1 0.1 0.1
X V V 0 0 0 0 0 0 0 0 0 0 0 0
S S_X S_X 0 0 0 0 1.0 0 0 0 0 0 0 0
V años years 0.3 0.2 -1 -1 0 0 0 0 0 -1 0.2 0.1
X V V 0 0 0 0 0 0 0 0 0 0 0 0
S S_X S_X 0 0 0 0 1.0 0 0 0 0 0 0 0
V . . 0.1 0 -1 -1 0 0 0 0 0 -1 1.4 1.4
```

```
X V V 0 0 0 0 0 0 0 0 0 0 0 0
S S_X S_X 0 0 0 0 1.0 0 0 0 0 0 0 0
X </s> </s> 0 0 -1 -1 0 0 0 -1 0 0 0 0
S S_X S_X 0 0 0 0 1.0 0 0 0 0 0 0 0
```

This tree and this rule sequence are two representations of the same translation hypothesis. Please make sure that you understand the relationship between them before starting the practical. Contact the practical demonstrator if this is not clear.

2 Practical Exercise

The following sections describe a set of subtasks to be done during the practical, and raise questions which you must answer in your practical report. Please seek help from the demonstrator should you encounter problems with this part of the practical.

2.1 First part

Two translation grammars have been created from a parallel corpus. They are provided in the directories `$DIR/rules/test30/A/` and `$DIR/rules/test30/B/`, respectively. In each directory, there is a sentence-specific rulefile to be used for each sentence; so you can see 30 files in each directory.

1. Translate the 30 sentences with each grammar and score the output translations against their English reference translation. Which grammar obtains the better BLEU score? Which of the two runs was faster?

   Note: the decoder outputs a translation lattice for each input sentence. In order to evaluate all sentences (using `ScoreBLEU.sh` from practical 1), first you will need to print the 1-best translation for all sentences into a single file, as follows:

   ```
   PrintTranslation.sh -d directory > output_file
   ```

   where the directory contains the 30 output translation lattices.

2. According to sentence-level BLEU scores, is every translated sentence better with grammar B than with A?

   (a) Examine two sentences where you obtain a significantly better score with rulefile B, showing the input sentence, the English reference and the two alternative translations. Do you think the sentence-level BLEU score reflects a true improvement in translation quality? Show 5 examples of clear improvement in the produced English hypothesis.

   (b) Repeat the previous question with a sentence that gets lower score with rulefile B. Does the BLEU score reflect a true degradation in translation quality? Why do you think it is worse?

3. Compare rulefiles A and B for sentence 27. What are the main differences you observe? Pay special attention to the nonterminals used in the various columns. How do these differences in the rulefiles explain the differences in the produced translation?

4. Give further support to your previous answer by drawing the 1-best derivation tree for sentence 27 when translated by each ruleset. Which rule sequence is used in each case?

5. Now align the 30 sentences towards their respective English reference. In order to do that, you will need to retranslate the input using the `--referencefilter.load` option. This forces the translation system to produce all possible derivations that can generate each English reference for each sentence[2]. By examining the resulting output transducers, determine for how many input sentences can the reference be generated? Compare this for grammar A and B.

6. Is there any sentence where the reference can be generated by grammar A but NOT by B? Justify your answer comparing one of the rulefiles.

7. To conclude, which translation grammar is more expressive? Summarise the main properties of each grammar, and name them appropriately.

2.2 Second part

We would like to assess the expressivity of a third translation grammar, which can be found in `$DIR/rules/C/`.

1. Compare rulefiles B and C for sentence 27. How many rules do we have in each case? What differences do you observe in the rules?

2. Align the 30 sentences towards their English reference with grammar C. **Please make sure you do not omit the `--referencefilter.load` option**. For how many input sentences can the reference be generated in this case? Are there references that can be generated by grammar B but NOT by C, and viceversa?
   N.B. Do not use the `--hifst.alilatsmode` option for this computation.

3. Take two sentences that can *only* be aligned by grammar C, and draw their 1-best derivation tree. Explain why the tree cannot be generated by grammar B, detailing which rule nesting cannot be performed. Justify your choice by comparing the ruleset files.
   N.B Make sure to use the `--hifst.alilatsmode` option for this computation.
   There is a known error in processing sentences 9 and 22 using this version of HiFST. You can skip these two sentences for this question.

[2]Note that you should use the sentence-specific English references encoded as FSTs in `$DIR/reference/test30.fst/r.?.eng.idx.fst.gz`.