NYUAD CS-AD 220 – Spring 2016
Natural Language Processing

Assignment #2
Finite State Machines
Assigned Feb 18, 2016

Due Mar 10, 2016 (11:59pm)

I. Grading & Submission

This assignment is about the development of finite state machines using the OpenFST and Thrax toolkits. The assignment accounts for 15% of the full grade. It consists of three exercises. The first is a simple "machine translation" system for animal sounds to help with learning the tools. The second is about modeling how numbers are read in English and French. And the third is about Spanish verb conjugation. The answers should be placed in a zipped folder with separate sub-directories for each exercise.

The assignment is due on March 10 before midnight (11:59pm). For late submissions, 10% will be deducted from the homework grade for any portion of each late day. The student should upload the answers in a single zipped to NYU Classes (Assignment #2).

II. Before Starting

You need to install the following three tools: OpenFST, Thrax and Graphviz. You also need to download the files associated with this assignment. PLEASE do this ASAP and ask for help if you have a problem setting up. Although this assignment will guide you through some of the basic commands from these tools, you should read through the websites of these tools (specifically Thrax and OpenFST, but less so, for Graphviz) to help you with the assignment.

A. OpenFST

OpenFST is a library for constructing, combining, optimizing, and searching weighted finite-state transducers (FSTs). Open FST Version 1.4.1 was used in preparing this assignment.

Download from http://www.openfst.org/twiki/bin/view/FST/FstDownload.
Unpack (unzip and untar) the downloaded gzipped tarball.
Go in the main directory and install OpenFST with the following three commands:

```
./configure --enable-far --enable-pdt --enable-mpdt
make
make install
```

If successful, the fst* commands will be available to you in /usr/local/bin/. They should also be accessible from anywhere. Test with this command:

```
fstprint --help
```

Start learning about OpenFST by taking the Quick Tour linked under: http://www.openfst.org/twiki/bin/view/FST/

B. Thrax

The OpenGrm Thrax tools compile grammars expressed as regular expressions and context-dependent rewrite rules into weighted finite-state transducers. It makes use of functionality in the OpenFST library to create, access and manipulate compiled grammars. Version 1.1.0 was used in preparing this assignment.

Download from http://www.openfst.org/twiki/bin/view/GRM/ThraxDownload
Unpack (unzip and untar) the downloaded gzipped tarball.
Go in the main directory and install with the following three commands:

```
./configure
make
make install
```

If successful, the thrax* commands will be available to you in /usr/local/bin/. They should also be accessible from anywhere. Test with this command:

```
thraxmakedep --help
```

C. Graphviz

The Graphviz layout programs take descriptions of graphs in a simple text language, and make diagrams in useful formats, such PDF. Download from http://www.graphviz.org (follow the relevant instructions to your OS).

If successful, the **dot** command will be available to you in /usr/local/bin/. It should also be accessible from anywhere. Test with this command:

```
dot -?
```

D. Assignment Files

The assignment files are downloadable in a zipped directory named CS-AD-220-S2016-Assignment-2 and containing the following:

```
NYUAD-CS-AD-220-S2016-Assignment-2.pdf    #This file
testFAR.pl                    #Code to test of FST archives (FAR)
evalFAR.pl                    #Code to evaluate FARs
FarmTrans                     #EX1: Farm Machine Translation
      farm.grm                      #Grammar file
      farm.dict                     #Dictionary
      farm.en2ar.examples           #Evaluation file
      farm.ar2en.examples           #Evaluation file
Numbers                       #EX2: Counting in English and French
      EnglishNumbers.grm            #Grammar file
      EnglishNumbers.examples       #Evaluation file
      FrenchNumbers.examples        #Evaluation file
SpanishVerbs                  #EX2: Spanish Verb Conjugation
      SpanishVerbs.grm              #Grammar file
      Spanish.lex                   #Lexicon file
      SpanishVerbs.examples         #Evaluation file
```

III. Exercises

EX1. Farm Machine Translation (20 points)

Our first exercise is about translating the language different animal sounds from English to Arabic:

Animal	English	Arabic
Sheep	baa+!	mAA+!
Cat	meo+w!	myA+w!
Dog	woo+f!	Eaw+!
Cow	moo+!	muww+!

The folder FarmTrans contains a basic Thrax grm file (farm.grm) that specifies the form of English Sheep language and Arabic Sheep language; as well as, loading a small dictionary that maps the alphabets of the languages to each other. The file is documented.

To build the farm.far (FAR=FST Archive), apply the following commands:

```
thraxmakedep --save_symbols farm.grm
make
```

To test the created FARMTRANS_en2ar Finite State Machine, call:

```
perl ../testFAR.pl farm.far FARMTRANS_en2ar
```

Provide input from standard input. Input "baaa!" produces (input <tab> output):

```
baaa! mAAA!
```

But "ba" (invalid Sheep language) gets <REJECT>.

Now, to view the FARMTRANS_en2ar FST, we need first to extract it from the FAR (FST archive) file:

```
farextract --filename_suffix=.fst farm.far
```

You can view the machine in text format:

```
fstprint FARMTRANS_en2ar.fst
```

The format produced by fstprint is

```
<start-stat><tab><end-state><tab><input><tab><output>
```

Accepting states appear on a line by themselves.

You can draw it to a pdf file:

```
fstdraw FARMTRANS_en2ar.fst |dot -Tpdf > FARMTRANS_en2ar.pdf
```

(if you have a problem with this command; try the following instead to generate in .ps format, and use another tool such as preview or ps2pdf to convert to pdf:

```
fstdraw FARMTRANS_en2ar.fst |dot -Tps > FARMTRANS_en2ar.ps
```

Your Mission:
Extend the FARMTRANS_en2ar machine to handle Cat, Dog and Cow in addition to Sheep. The machine should also handle multiple utterances of animals in the same line: e.g.

```
        baaa!moooo!woof!woof!meooooooow!
→       mAAA!muwww!Eaw!Eaw!myAAAAAAw!
```

To help you test your machine, we provided a set of examples with correct answers (`farm.en2ar.examples`). To test coverage of phenomena, do:

```
perl ../evalFAR.pl farm.far FARMTRANS_en2ar farm.en2ar.examples
```

This script evaluates how you are doing and returns an accuracy score. The provided system gets 60% accuracy. You need to reach 100%.

Export a second machine FARMTRANS_ar2en that translates in the other direction. This should be done with a single additional command. Evaluate with `farm.ar2en.examples`.

Deliverables:
A directory containing the new farm.grm and farm.dict files. Include also a new farm.far file which reaches 100% accuracy when evaluated on the examples files (en2ar and ar2en). Finally, include a PDF drawing of the final FARMTRANS_en2ar.fst file.

EX2. English and French Numbers (40pts)

In this part of the assignment, you will build a FST that maps digits to words: e.g.,

`28193 → twenty–eight thousand one hundred and ninety–three`

In the directory Numbers, you will find a simple English number grammar (EnglishNumbers.grm) that maps digits from 0 to 999. You will use it as an example to build two new grammars:

- EnglishNumbers-1M.grm which models digit-to-word mapping from 0 to 1000000 (zero to one million). **(20pts)**
- FrenchNumbers.grm which models French digit-to-word mappings from 0 to 100. French numbers can be tricky to model. Consider for instance the number 97 which is read as 97 *quatre-vingt-dix-sept* (lit. *four-twenty-ten-seven*). See Appendix 1 for all these numbers. **(20pts)**

You will also find in the directory two examples files for English and French numbers which will help you evaluate your models.

Deliverables
A directory with EnglishNumbers-1M.grm and FrenchNumbers.grm; the .far version of them. The two should score 100% accuracy on the example files.

EX3. Spanish Verbs (40pts)

In this part of the assignment, you will build a FST that conjugates Spanish Verbs: e.g., for an input specifying the Spanish verb 'to talk' hablar with First Person Indicative features in the present tense: *hablar+Ind1S*, the output is *hablo*. An inverse machine will analyze the verb *hablo* into *hablar+Ind1S*. Muy fácil , ¿no?

In the SpanishVerbs directory, you will find a basic Spanish Verb grammar which you can compile into an FST archive as done with other parts of the assignment. You will also find a small incomplete lexicon, as well as a set of examples to help you evaluate your progress.

Some facts to consider for this exercise:

1. Spanish verbs are grouped in three classes depending on the verbal ending in the infinitive (which happens to be also the citation form): -ar, -er, and –ir verbs. These classes vary in the form of their inflected (conjugated) verbs, although not always: compare habl**ar**-habl**as**-habl**amos**, cog**er**-cog**es**-cog**emos**, and viv**ir**-viv**es**-viv**imos**. We will only consider eight verbs for a subset of the verbal paradigm. The verbs are listed in Appendix 2.
2. Some verbs involve a regular c-z mutation:
   ven**c**er-ven**z**o-ven**c**es-ven**z**a-ven**z**an
   cru**z**ar-cru**z**o-cru**z**as-cru**c**e-cru**c**en
3. Some verbs with the letter g have a couple of mutations (g-j and g-gu):
   co**g**er-co**j**o-co**g**es-co**j**amos
   pa**g**ar-pa**g**o-pa**g**as-pa**gu**emos

Deliverables

A directory with a new SpanishVerbs.grm and Spanish Lexicon; a compiled SpanishVerb machine that scores 100% accuracy on the example files.

Appendix 1: French Numbers 0-100

0 zéro
1 un
2 deux
3 trois
4 quatre
5 cinq
6 six
7 sept
8 huit
9 neuf
10 dix
11 onze
12 douze
13 treize
14 quatorze
15 quinze
16 seize
17 dix-sept
18 dix-huit
19 dix-neuf
20 vingt
21 vingt et un
22 vingt-deux
23 vingt-trois
24 vingt-quatre
25 vingt-cinq
26 vingt-six
27 vingt-sept
28 vingt-huit
29 vingt-neuf
30 trente
31 trente et un
32 trente-deux
33 trente-trois
34 trente-quatre
35 trente-cinq
36 trente-six
37 trente-sept
38 trente-huit
39 trente-neuf
40 quarante
41 quarante et un
42 quarante-deux
43 quarante-trois
44 quarante-quatre
45 quarante-cinq
46 quarante-six
47 quarante-sept
48 quarante-huit
49 quarante-neuf
50 cinquante
51 cinquante et un
52 cinquante-deux
53 cinquante-trois
54 cinquante-quatre
55 cinquante-cinq
56 cinquante-six
57 cinquante-sept
58 cinquante-huit
59 cinquante-neuf
60 soixante
61 soixante et un
62 soixante-deux
63 soixante-trois
64 soixante-quatre
65 soixante-cinq
66 soixante-six
67 soixante-sept
68 soixante-huit
69 soixante-neuf
70 soixante-dix
71 soixante et onze
72 soixante-douze
73 soixante-treize
74 soixante-quatorze
75 soixante-quinze
76 soixante-seize
77 soixante-dix-sept
78 soixante-dix-huit
79 soixante-dix-neuf
80 quatre-vingts
81 quatre-vingt-un
82 quatre-vingt-deux
83 quatre-vingt-trois
84 quatre-vingt-quatre
85 quatre-vingt-cinq
86 quatre-vingt-six
87 quatre-vingt-sept
88 quatre-vingt-huit
89 quatre-vingt-neuf
90 quatre-vingt-dix
91 quatre-vingt-onze
92 quatre-vingt-douze
93 quatre-vingt-treize
94 quatre-vingt-quatorze
95 quatre-vingt-quinze
96 quatre-vingt-seize
97 quatre-vingt-dix-sept
98 quatre-vingt-dix-huit
99 quatre-vingt-dix-neuf
100 cent

Appendix 2: Eight Spanish Verbs

The table below specifies a part of the conjugation of eight Spanish verbs.

Spanish	**hablar**	**cruzar**	**llegar**	**pagar**	**vencer**	**coger**	**vivir**	**partir**
English	**talk**	**cross**	**arrive**	**pay**	**overcome**	**grab**	**live**	**leave**
Inf	hablar	cruzar	llegar	pagar	vencer	coger	vivir	partir
Ind1S	hablo	cruzo	llego	pago	venzo	cojo	vivo	parto
Ind2S	hablas	cruzas	llegas	pagas	vences	coges	vives	partes
Ind3S	habla	cruza	llega	paga	vence	coge	vive	parte
Ind1P	hablamos	cruzamos	llegamos	pagamos	vencemos	cogemos	vivimos	partimos
Ind3P	hablan	cruzan	llegan	pagan	vencen	cogen	viven	parten
Sub1S	hable	cruce	llegue	pague	venza	coja	viva	parta
Sub2S	hables	cruces	llegues	pagues	venzas	cojas	vivas	partas
Sub3S	hable	cruce	llegue	pague	venza	coja	viva	parta
Sub1P	hablemos	crucemos	lleguemos	paguemos	venzamos	cojamos	vivamos	partamos
Sub3P	hablen	crucen	lleguen	paguen	venzan	cojan	vivan	Partan

The rows correspond to the citation form in Spanish, English gloss, infinitive (Inf) form, and the Present Indicative (Ind*) and Present Subjunctive (Sub*) forms, both conjugated for different persons (1, 2, 3) and numbers (S=Singular, P= Plural). Parts of the verbal paradigm and other tenses are ignored to keep the problem focused for this assignment.