Evaluating Speech Recognition

CS 136a Lecture 9

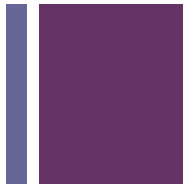
February 25, 2020

Professor Meteer

Testing

- All software requires regression testing
  - Develop tests that capture both standard use cases and edge use cases
  - At every release additional tests are developed to ensure new features work
  - The software is also run through the original battery of tests to ensure new feature don't interfere with how previous features work
  - Tests are generally totally automated
- Testing user interfaces require the same regression testing
  - Challenge: you are testing a "path", which may be different depending on how previous steps worked
  - How to create a regression test that isn't just trying things out

Two ways to Evaluate



- Intrinsic Methods
  - Transcription Accuracy
    - Word Error Rate
    - Automatic methods, toolkits
    - Limitations
  - Concept Accuracy
    - Limitations
- Extrinsic Methods
  - Cheap (but not systematic)
    - Put the grammar in an application
    - Deploy & see if people keep using it
  - The right way (but can be expensive)
    - Identify a set of test users
    - Track actions & analyze

Component Evaluation

- How to evaluate the ‘goodness’ of a word string output by a speech recognizer?
- Terms:
  - ASR hypothesis: ASR output
  - Reference transcription: ground truth – what was actually said

Transcription Accuracy

- Word Error Rate (WER)
  - Minimum Edit Distance: Distance in words between the ASR hypothesis and the reference transcription
    - Edit Distance: = (Substitutions+Insertions+Deletions)/N
    - For ASR, usually all weighted equally but different weights can be used to minimize difference types of errors
  - WER = Edit Distance * 100
  - Applying ”minimum edit distance” to speech

    It’s easy to recognizer speech

    It’s easy to wreck a nice beach
  - What’s the “edit distance”?

Other Types of Error Analysis

- What speakers are most often misrecognized (Doddington '98)
  - Sheep: speakers who are easily recognized
  - Goats: speakers who are really hard to recognize
  - Lambs: speakers who are easily impersonated
  - Wolves: speakers who are good at impersonating others
- What sounds (context-dependent phones) are least well recognized?
  - Can we predict this?
- What words are most confusable (confusability matrix)?
  - Can we predict this?

SCLite

- Program developed by NIST to score speech recognition competitions
- First run a speech recognizer on a set of audio files
- Input to SCLite
  - ".ref" file with the actual transcriptions (one per line)
  - ".hyp" file with the recognizers output (one per line)
- Output
  - Overall score (accuracy, substitutions, deletions, insertions)
  - Score by speaker (needs special file naming conventions)
  - Sentence by sentence errors
  - Summary of errors (how many of each substitution type, how often each word was deleted, inserted …)

Performance: results.sys

SYSTEM SUMMARY PERCENTAGES by SPEAKER

/home/g/grad/lvweber/Desktop/final.trn

SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
s01	3	15	86.7	6.7	6.7	6.7	20.0	66.7
s02	3	15	60.0	13.3	26.7	0.0	40.0	100.0
s03	8	74	70.3	23.0	6.8	0.0	29.7	100.0
s04	5	38	65.8	23.7	10.5	0.0	34.2	100.0
s05	10	108	75.9	20.4	3.7	0.9	25.0	70.0
s06	9	75	66.7	22.7	10.7	5.3	38.7	100.0
s07	9	107	89.7	8.4	1.9	0.0	10.3	100.0
s08	5	37	70.3	27.0	2.7	2.7	32.4	100.0
Sum/Avg	52	469	75.3	18.6	6.2	1.5	26.2	92.3
Mean	6.5	58.6	73.2	18.1	8.7	2.0	28.8	92.1
S.D.	2.8	37.8	10.4	7.6	8.0	2.7	10.0	14.7
Median	6.5	56.0	70.3	21.5	6.7	0.5	31.1	100.0

Evaluating Performance

- Word Error Rate =

$$100 * \frac{\text{(Insertions + Substitutions + Deletions)}}{\text{Total Words in Correct Transcript}}$$

(note: WER can be > 100%)

Alignment example from .pra file

REF: portable **** PHONE UPSTAIRS last night so

HYP: portable FORM OF STORES last night so

Eval I S S

WER = 100 (1+2+0)/6 = 50%

Thanks to Dan Jurafsky for these slides

NIST sctk-1.3 scoring software: Computing WER with sclite

- http://www.nist.gov/speech/tools/
- Sclite aligns a hypothesized text (HYP) (from the recognizer) with a correct or reference text (REF) (human transcribed)

```
id: (2347-b-013)

Scores: (#C #S #D #I) 9 3 1 2

REF:  was an engineer SO I   i was always with ****
  **** MEN UM   and they

HYP:  was an engineer ** AND i was always with THEM
  THEY ALL THAT and they

Eval:                  D  S                       I
  I    S   s
```

Thanks to Dan Jurafsky for these slides

Sclite output for error analysis: .dtl file

```
CONFUSION PAIRS              Total              (972)

With >=  1 occurances (972)

 1:     6  ->  (%pause) ==> on
 2:     6  ->  the ==> that
 3:     5  ->  but ==> that
 4:     4  ->  a ==> the
 5:     4  ->  four ==> for
 6:     4  ->  in ==> and
 7:     4  ->  there ==> that
 8:     3  ->  (%pause) ==> and
 9:     3  ->  (%pause) ==> the
10:     3  ->  (a-) ==> i
11:     3  ->  and ==> i
12:     3  ->  and ==> in
13:     3  ->  are ==> there
14:     3  ->  as ==> is
15:     3  ->  have ==> that
16:     3  ->  is ==> this
17:     3  ->  it ==> that
18:     3  ->  mouse ==> most
19:     3  ->  was ==> is
20:     3  ->  was ==> this
21:     3  ->  you ==> we
22:     2  ->  (%pause) ==> it
23:     2  ->  (%pause) ==> that
24:     2  ->  (%pause) ==> to
25:     2  ->  (%pause) ==> yeah
26:     2  ->  a ==> all
27:     2  ->  a ==> know
28:     2  ->  a ==> you
29:     2  ->  along ==> well
30:     2  ->  and ==> it
31:     2  ->  and ==> we
32:     2  ->  and ==> you
33:     2  ->  are ==> i
34:     2  ->  are ==> were
```

Thanks to Dan Jurafsky for these slides

Naming conventions

- SCLite assumes audio files and the utterances in the .ref and .hyp files follow specific naming conventions

  SPEAKER_TEST_<digit>

- .ref and .hyp files use this convention to label each utterance using SNOR format
  - Text (SPEAKER_TEST_<digit>)

- Examples
  - .ref

    Hi let me have a small spinach and feta pizza with bacon and diced tomatoes please (LDThorne_001)

    Hi can I get two small cheese pizzas please (LDThorne_003)

    I want a small Wisconsin six cheese pizza with pepperoni (LDThorne_005)
  - .hyp

    hi Let me have a small spinach and diced tomatoes please (LDThorne_001)

    Hi Can I get two small cheese pizzas please (LDThorne_003)

    i want a Small Extra Cheese pizza (LDThorne_005)

Are there better metrics than WER?

- WER useful to compute transcription accuracy
- But should we be more concerned with meaning ("semantic error rate")?
  - Good idea, but hard to agree on approach
  - Applied mostly in spoken dialogue systems, where semantics desired is clear
  - What ASR applications will be different?
    - Speech-to-speech translation?
    - Medical dictation systems?

Concept Accuracy

- Spoken Dialogue Systems often based on recognition of Domain Concepts
- Input: I want to go to Boston from Baltimore on September 29.
- Goal: Maximize concept accuracy (total number of domain concepts in reference transcription of user input)

Concept	Value
Source City	Baltimore
Target City	Boston
Travel Date	Sept. 29

+ Concept Accuracy vs. WER

- CA Score: How many domain concepts were correctly recognized of total N mentioned in reference transcription
  - Reference: I want to go from Boston to Baltimore on September 29
  - Hypothesis: Go from Boston to Baltimore on December 29
  - 2 concepts correctly recognized/3 concepts in ref transcription * 100 = 66% Concept Accuracy
- What is the WER?
  - 3 Ins+2 Subst+0Del/11 * 100 = 45% WER (55% Word Accuracy)

Sentence Error Rate

- Sentence Error Rate
  - Percentage of sentences with at least one error
    - Transcription error
    - Concept error
- Which Metric is Better?
  - Transcription accuracy?
  - Semantic accuracy?

+ Evaluating speech in Alexa

- Need to have access to the history
  - Single history for all devices on the account
  - Need to transform that into file format for evaluation
- Steps to avoid hand cleaning
  - Open history on the web, copy and paste utterances into an editor

  Alexa Today at 10:13 AM on Arlington Livingroom Echo Dot
  alexa what's the weatherToday at 10:05 AM on Marie's Echo Dot
  alexaToday at 10:05 AM on Marie's 4th Echo
  alexa what time is itToday at 8:31 AM on Marie's 4th Echo
  play w. b. u. r.Today at 7:37 AM on Arlington Livingroom Echo Dot
  alexaToday at 7:37 AM on Arlington Livingroom Echo Dot
- Goal:
  - Grouped by source (e.g. which group the utterances belong to)
  - Ordered by time
  - Without the “alexa” start word

+ Cleaning Alexa History Data

- Review the format

  Off Today at 8:39 **AM on** Arlington Livingroom Echo Dot

  Alexa Today at 8:38 **AM on** Arlington Livingroom Echo Dot

  alexa what time is it Today at 8:31 **AM on** Marie's 4th Echo

  what's the weather tomorrow Yesterday at 11:25 **PM on** Marie's Echo Dot

  Alexa Yesterday at 11:25 **PM on** Marie's Echo Dot

- Need: Utterance, time, source

- Requirements
  - Remove ”alexa”: Text editor with “replace”
  - Remove unnecessary words: “Today at”, “AM on”
  - Sort so that all the utterances from the same device and in order of time

Running SCLite

- Direct call

```
sclite –r  results.ref   -h results.hyp  -i rm –O results_dir/  -o all

sclite –r  results.ref   -h results.hyp  -i rm –O results_dir/  -o dtl
```

  - DTL output shows details on substitutions, deletions and insertions

Final steps

- Excel

off	8:39	Arlington_Livingroom_Echo_Dot	1
play w. b. u. r.	7:37	Arlington_Livingroom_Echo_Dot	2
what time is it	8:31	Marie's_4th_Echo	1
off	6:39	Marie's_Echo_Dot	1
off	5:52	Marie's_Echo_Dot	2
snooze	6:30	Marie's_Echo_Dot	3

- Concatenate

off (Arlington_Livingroom_Echo_Dot_1)

play w. b. u. r. (Arlington_Livingroom_Echo_Dot_2)

what time is it (Marie's_4th_Echo_1)

off (Marie's_Echo_Dot_1)

off (Marie's_Echo_Dot_2)

snooze (Marie's_Echo_Dot_3)

what's the weather (Marie's_Echo_Dot_4)

what's the weather tomorrow (Marie's_Echo_Dot_5)

Creating the .ref file

- Transcribe your utterances (wav files)

I would like a small cheese pizza (YOURNAME_001)

I would like two large chicken pizzas (YOURNAME_002)

I would like three medium cheese pizzas please (YOURNAME_003)

I would like one large cheese pizza and one large pepperoni pizza (YOURNAME_004)

I want one medium pepperoni and sausage pizza (YOURNAME_005)

Can I get um one medium spinach pizza please (YOURNAME_006)

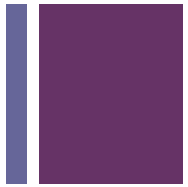
I want one medium pepperoni and sausage pizza and one small mushroom pizza (YOURNAME_007)

Can I get one large pizza with pepperoni please (YOURNAME_008)

I want two small pizzas with sausage and one small pizza with mushrooms (YOURNAME_009)

I would like um five medium pizzas with sliced italian sausage (YOURNAME_010)

Creating the .hyp file



Loop through the directory of .emma files

```
while (<INFILE>) {
  chomp;
  if (/"hypothesis":\s+"(.*)"/) {   #this will be different for emma
$hyp = $1;
print OUTHYP "$hyp ($fname)\n";
next;
  }
}
```

Method

- Text editor with an easy way to do global replace
- Turn it into csv format
- Read into excel
- Sort
  - First on text so empty utterances can be deleted
  - Next on device, then time
- Create the final version: SNOR format
  - First, get rid of spaces in device name
  - Number sequentially within a device
  - Concatenate

Fixing the audio

- SOX: The Swiss Army knife of audio processing
  - Available through Sourceforce here:
    - http://sourceforge.net/projects/sox/files/sox/
  - Copy it into /Applications/ and double click on the compressed file (if it didn't open into a directory by itself). Set the path environment variable from the terminal command line:
    - export PATH=$PATH:/Applications/sox-14.4.1/

Using Sox

- Get information about the file

  soxi 001.wav

  ```
  Input File     : '001.wav'
  Channels       : 2
  Sample Rate    : 44100
  Precision      : 16-bit
  Duration       : 00:00:02.46 = 108544 samples = 184.599 CDDA sectors
  File Size      : 434k
  Bit Rate       : 1.41M
  Sample Encoding: 16-bit Signed Integer PCM
  ```

- Change the file

  sox 001.wav -r 8000 0015.wav

- Resulting file

  soxi 0015.wav

  ```
  Input File     : '0015.wav'
  Channels       : 2
  Sample Rate    : 8000
  Precision      : 16-bit
  …
  ```

Operating in a batch

```
#!/usr/bin/perl –w
$audio_dir = shift@ARGV
opendir(DIR,$audio_dir) || die "Can't open $audio_dir";
local(@filenames) = readdir(DIR);
closedir(DIR);

$output_dir = shift@ARGV;    #output directory
print "Input: $audio_dir Output: output_dir\n";

for $file (@filenames) {
  if ($file =~ /\.wav/) {
    $wavfile = $audio_dir . $file;
    $file =~ s/wav/emma/;
    $outfile = $output_dir . $file;
    print "Processing $wavfile to $outfile\n";
      system("bash scripts/call_reco.sh $wavfile $outfile");
 }}
```