NST Meeting

April 2012

Marcus Tomalin & Bill Byrne
Cambridge University Engineering Department
Natural ASR

ASR output contains errors, from speakers and ASR systems:

- NO ONE QUESTIONS THAT CLINTON NEEDS AND OFF
- IS THIS BOMBING SCENE IS AN ESPECIALLY SOPHISTICATED OPERATION
- HOWARD DEAN ESPECIALLY WELCOME THAT POSSIBILITY
- WENT YOU ABLE TO PULL ANYTHING OUT

If ‘naturalness’ = grammaticality and acceptability, then ASR often ‘unnatural’:

- Grammaticality: a grammatical sentence is formed according to a grammar
- Acceptability: characterises a native speaker’s intuitions about linguistic data

Both words are problematical:

- grammaticality/acceptability judgements often informal (linguist + close friends)
- degrees of grammaticality/acceptability indicated by idiosyncratic means (e.g., *, **, ?, ??)
- grammaticality relates to competence (I-language); idealised knowledge
- acceptability relates to performance (E-language); context, psychology, pragmatics, etc
Natural ASR

Grammaticality/acceptability degraded by ASR:

- speaker/hearer creates grammatical sentences (I-language)
- actual speech data contains slips, hesitations, restarts, etc (E-language)
- ASR systems add further errors

<table>
<thead>
<tr>
<th>I-language:</th>
<th>What do these cases have in common?</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-language:</td>
<td>What do these case have in common?</td>
</tr>
<tr>
<td>ASR:</td>
<td>WHAT TO THESE CASE HAVE IN COMMON?</td>
</tr>
</tbody>
</table>

A set-theoretical framework for a given corpus:

- if $G = \{\gamma : \gamma$ is a grammatical sentence$\}$,
- if $A = \{\alpha : \alpha$ is a acceptable sentence$\}$,
- then $N = G \cap A$, where $N$ is the set of ‘natural’ sentences

The task is to make ASR output more ‘natural’...
Natural ASR

At NST meeting in January, a training/testing corpus defined:

- extract all grammatical + acceptable REF sentences
- extract corresponding subset of HYP sentences
- ‘unnatural’ HYPs can be compared to corresponding ‘natural’ REFs

Two ASR data sets prepared:

- **EARS BN-E eval[03,04] and dev04 testsets**: c.300 ‘natural’ REFs [16% of all REFs]
- **EARS SW-E eval[03,04] and dev04 testsets**: c.2K ‘natural’ REFs [17% of all REFs]

Also, four basic operations to improve ‘naturalness’ of HYP segs were identified:

- **permutation**: rearrange token order: she running is → she is running
- **insertion**: insert tokens: she running → she is running
- **deletion**: delete tokens: she is was running → she is running
- **substitution**: replace tokens: she his running → she is running
Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG) – a lexicalised grammar formalism:
Can implement permutation, insertion, and deletion in CCG-based framework

- developed by Ajdukiewicz (1935) and Bar-Hillel (1953)
  - see Tomalin 2006, 67-73, Steedman 2000
- grammatical constituents associated with syntactic types (categories)
- syntactic rules apply depending on the category of the input constituents
- constituents are identified by their categories as being either primitives or functions:
  - man = N (a primitive)
  - the = NP/N (result/argument)
- ‘/’ means ‘rightward combining’, ‘\’ means leftward combining:
  - sings = S\NP, intransitive verb (a functor) ‘NP sings’
- functional application rules, functional combination rules, type-raising rules specified

Combinatory rules enable functors to combine with arguments...
Combinatory Categorial Grammar

Functional Application Rules:

1. $X/Y, Y \rightarrow X \langle\rangle$ [forward application]
2. $Y, X\backslash Y \rightarrow X \langle\rangle$ [backward application]

Semantic representations can be included using lambda calculus

CCG Training/Test Corpus:

- **CCGbank** = 99.4% of c.50K sentences in Penn Wall Street Journal Treebank corpus (sections 00-24)
Combinatory Categorial Grammar

A CCG system developed by Yue Zhang and Stephen Clark (CU Computer Lab)
Generates grammatical sentences from unordered input token sequence

- extract CCG rules from CCGBank corpus (training data = sections 02-21)
- POS-tag input tokens and generate HYPs using CCG rules [bottom-up, recursive]
- best-first algorithm reduces search space; also a ‘timeout’ constraint (300 secs)
- search for optimal parse guided by large-margin training (using CCGbank data)
- output 1-best HYP with best score

Better BLEU scores than dependency-grammar for WSJ task (cf Wan 2009):

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency Grammar</td>
<td>33.7</td>
</tr>
<tr>
<td>CCG</td>
<td>46.1</td>
</tr>
</tbody>
</table>

CCGBank test set = section 23 (1913 sentences)
Combinatory Categorial Grammar

CCG-based system modified by adding an Ngram LM (LM):

- LM = 60k 4gram trained on 1B words of Gigaword data

The new steps:

- each partial CCG HYP (‘edge’) scored using syntax model and CCG-LM:
  \[ F(e) = f(e) + g(e), \]
  where \( F(e) \) is score for edge \( e \), \( f(e) \) is syntax model score, and \( g(e) \) is LM score

- output N-best HYPs
- rerank HYPs using LM

The CCG+LM system outperforms the CCG baseline:

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCG</td>
<td>43.2</td>
</tr>
<tr>
<td>CCG+CCG-LM</td>
<td>50.6</td>
</tr>
</tbody>
</table>

CCGBank test set = section 23 (1913 sentences)
Combinatory Categorial Grammar

CCG+LM framework helpfully constrains permutations:

- 1913 sentences in CCGBank test set
- avg. sentence length = 12.8 tokens
- total permutations for CCBank test set REFs = $1.2 \times 10^{23}$ (avg. 6B per sentence)
- total permutations for N-best CCG+LM HYPs = 488,406 (avg. 255 per sentence)

But it is overly-restrictive:

- 1404 (73%) of test set REF sequences don’t occur in respective N-best CCG+LM outputs
- CCG tools only output c.0.00001% of all possible hyps

Other problems with CCG-based approaches:

- generating grammatical sentences from unordered input is a harder task than improving ‘naturalness’ of highly ordered ASR output
- CCG framework is highly corpus-dependent – BLEU scores fall to 15-20% for non-WSJ tasks.
Evaluation Framework

Evaluating CCG+LM output is non-trivial: different metrics, different orderings

A specific example:

<table>
<thead>
<tr>
<th>CCG+LM HYPs</th>
<th>NG LogProb</th>
<th>WER</th>
<th>‘Naturalness’</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOU ARE PROBABLY A HYPOCRITE</td>
<td>-28.63</td>
<td>0.0</td>
<td>✔️</td>
</tr>
<tr>
<td>YOU PROBABLY ARE A HYPOCRITE</td>
<td>-28.64</td>
<td>40.0</td>
<td>✔️</td>
</tr>
<tr>
<td>PROBABLY YOU ARE A HYPOCRITE</td>
<td>-30.54</td>
<td>40.0</td>
<td>✔️</td>
</tr>
<tr>
<td>ARE YOU PROBABLY A HYPOCRITE</td>
<td>-31.66</td>
<td>40.0</td>
<td>✔️</td>
</tr>
<tr>
<td>YOU ARE A HYPOCRITE PROBABLY</td>
<td>-32.10</td>
<td>40.0</td>
<td>✔️</td>
</tr>
<tr>
<td>YOU ARE PROBABLY THE HYPOCRITE</td>
<td>-33.26</td>
<td>20.0</td>
<td>✔️</td>
</tr>
<tr>
<td>YOU PROBABLY ARE THE HYPOCRITE</td>
<td>-33.80</td>
<td>60.0</td>
<td>✔️</td>
</tr>
<tr>
<td>A HYPOCRITE YOU PROBABLY ARE</td>
<td>-34.73</td>
<td>60.0</td>
<td>✔️</td>
</tr>
<tr>
<td>ARE YOU A HYPOCRITE PROBABLY</td>
<td>-35.00</td>
<td>60.0</td>
<td>?</td>
</tr>
<tr>
<td>PROBABLY ARE YOU A HYPOCRITE</td>
<td>-35.22</td>
<td>60.0</td>
<td>?</td>
</tr>
</tbody>
</table>

Subject perceptual assessments are important...
Evaluation Framework

Web-based interface to obtain subjective responses:

This study assesses the 'naturality' (i.e., grammaticality/fluency/acceptability) of the output produced by Automatic Speech Recognition (ASR) and Statistical Machine Translation (SMT) systems. You will be asked to judge the naturality of sentences generated by such systems.

In each evaluation, you will be shown a sentence and asked to assess its degree of naturality using a scale from 1 (completely unnatural) to 5 (completely natural). Here are a few examples:

"THE LEFT THE COMPANY" score = 1 because it is a completely natural English sentence.
"PRESENT THE SAID HE AGREED" score = 2 because although part of it is natural (e.g., SAID HE AGREED) the rest is unnatural.
"THE AFTER MET FIRST A" score = 1 because the sentence is completely unnatural.

There are no right or wrong answers; you are not being tested. You simply need to respond as honestly as possible.

You should complete the entire evaluation. There are 18 sections and it should take about 20 minutes to complete them all. A response cannot be changed once it has been submitted. You should select your score from a drop-down list, and the instructions will be repeated at the start of each new section.

After you have completed all of the sections, you will be directed to a feedback questionnaire. This gives you an opportunity to tell us what you thought about this study. We would also like to gather some relevant demographic information, but all questions in the feedback questionnaire are optional.

If you have already registered, please log in instead.

To participate in this study you must be a native speaker of English, meaning that some dialect of English (e.g., British, American) is your first language, learnt from birth.

Email Address: 

Native language (from list)? Select from list.
Future Work

- develop full experimental framework for evaluating ‘naturalness’ improvement systems
- make CCG-LM performance more robust and less corpus-dependent
- enable CCG-based approaches to generate longer N-best lists
- utilise CCG information about sentence types (question, declarative, etc)
- process HYPs in larger discourse context (e.g., paragraphs not sentences):
  - **HYP**: Someone here is a **hypocrite**. You are probably the **hypocrite**.
  - anaphoric/cataphoric expressions, topic segmentation, etc
- process HYPs tokens at morphemic (rather than lexical) level:
  - **HYP**: Fred see Sue [+ -s] → Fred sees Sue
  - use morphological decomposition of training/test data
- learn mappings from HYPs to REFs using SMT finite state techniques (i.e., REF = language $x$, HYP = language $y$)
- use ASR lattices rather than N-best lists as input
- use insertion/permutation to insert filled pauses into TTS input text
References


