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., $\star$, $\star\star$, ?, ??)
- 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

I-language:	What	do	these	cases	have	in	common?
E-language:	What	do	these	case	have	in	common?
ASR:	WHAT	TO	THESE	CASE	HAVE	IN	COMMON?

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 permuation, 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 ($\rangle$) [forward application]
2. Y, X\Y $\rightarrow$ X ($\langle$) [backward application]

$$\begin{array}{ccc} \text{Fred} & \text{likes} & \text{Sue} \\ & & \\ \text{NP} & (\text{S}\backslash\text{NP})/\text{NP} & \text{NP} \\ \hline \multicolumn{3}{c}{\text{S}\backslash\text{NP}} \\ \hline \multicolumn{3}{c}{\text{S}} \end{array}$$

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):

System	**BLEU**
Dependecy Grammar	33.7
CCG	46.1

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:

System	BLEU
CCG	43.2
CCG+CCG-LM	50.6

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 = 1.2×10^{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

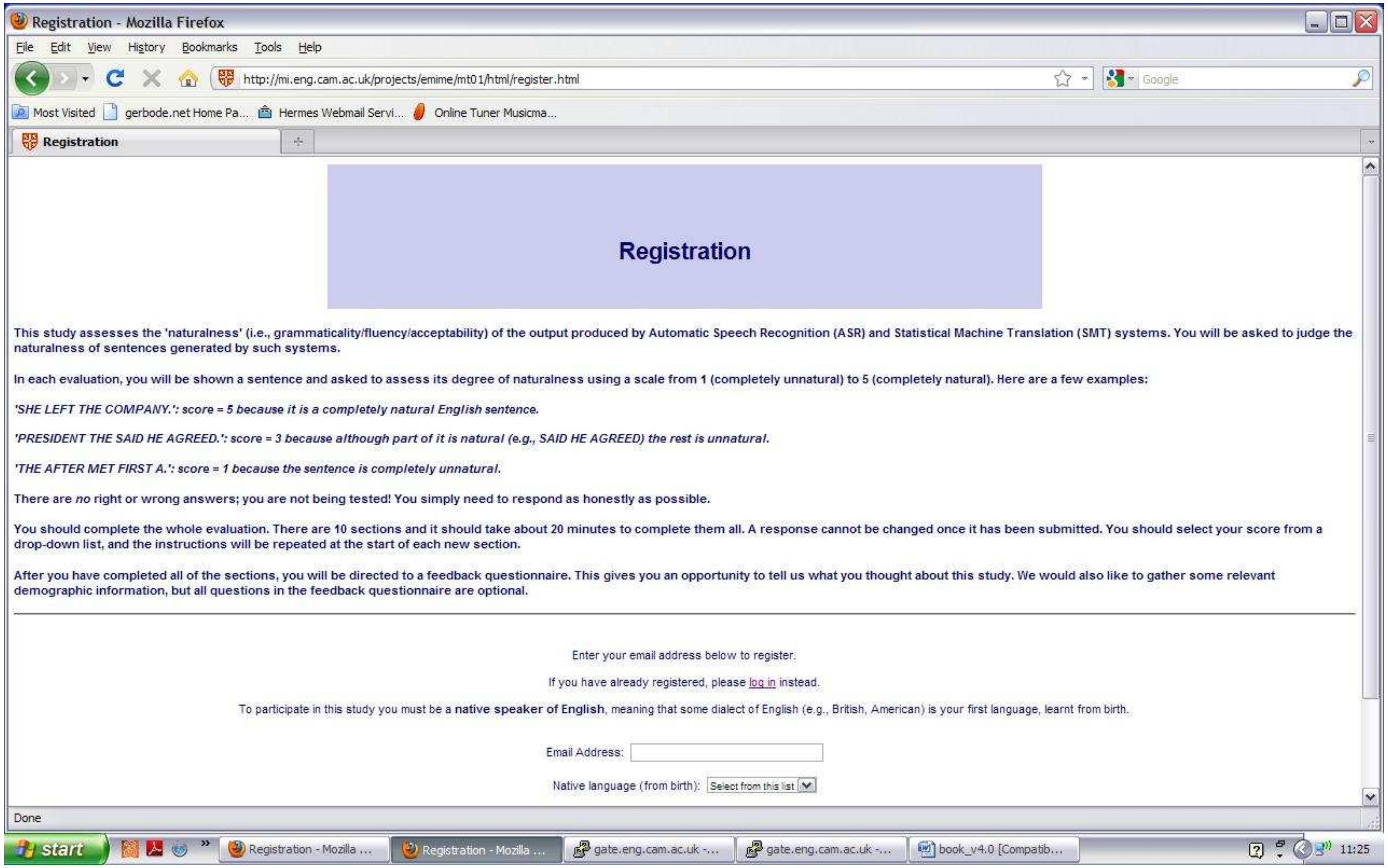
A specfic example:

CCG+LM HYPs	NG LogProb	WER	'Naturalness'
YOU ARE PROBABLY A HYPOCRITE	-28.63	0.0	[$\surd$]
YOU PROBABLY ARE A HYPOCRITE	-28.64	40.0	[$\surd$]
PROBABLY YOU ARE A HYPOCRITE	-30.54	40.0	[$\surd$]
ARE YOU PROBABLY A HYPOCRITE	-31.66	40.0	[$\surd$]
YOU ARE A HYPOCRITE PROBABLY	-32.10	40.0	[$\surd$]
YOU ARE PROBABLY THE HYPOCRITE	-33.26	20.0	[$\surd$]
YOU PROBABLY ARE THE HYPOCRITE	-33.80	60.0	[$\surd$]
A HYPOCRITE YOU PROBABLY ARE	-34.73	60.0	[$\surd$]
ARE YOU A HYPOCRITE PROBABLY	-35.00	60.0	[?]
PROBABLY ARE YOU A HYPOCRITE	-35.22	60.0	[?]

Subject perceptual assessments are important...

Evaluation Framework

Web-based interface to obtain subjective responses:



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] $\longrightarrow$ Fred sees Sue
  - use morphological decomposition of training/test data
- learn mappings from HYPs to REFs using SMT finite state techniques (i.e., REF = languge 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

- Steedman, M., *The Syntactic Process* (Cambridge, MA: The MIT Press, 2000)
- Tomalin, M., *Linguistics and the Formal Sciences* (Cambridge: CUP, 2006)
- Zhang, Y. and Clark, S., 'Syntax-based Grammaticality Improvement Using CCG and Guided Search', *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011
- Zhang, Y, Clark, S., and Blackwood, G., 'Syntax-based Word Ordering Incorporating a Large-scale Language Model', Proc. of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL-12), Avignon, 2012