Assessing and Improving the Performance of Speech Recognition for Incremental Systems

T. Baumann, M. Atterer, D. Schlangen

timo@ling.uni-potsdam.de

http://www.ling.uni-potsdam.de/~timo
Context:
Spoken Dialogue Systems
Context: Spoken Dialogue Systems

ASR

NLU

dialogue manager

history

domain

response generator

TTS

visual output

Spoken Dialogue Systems domain

ASR

response generator

NLU

TTS

visual output

dialogue manager

history

domain

context
Context: Spoken Dialogue Systems

- ASR
- NLU
- Dialogue Manager
- History
- Domain
- TTS
- Visual Output
- Response Generator
no reaction before the user finishes talking
Context: **Incremental**

Spoken Dialogue Systems

- **partial results** are being processed immediately
- reaction is quicker, back-channels are possible
Context: **Incremental**

Spoken Dialogue Systems

- **partial results** are being processed immediately
- reaction is quicker, back-channels are possible

My talk is about this arrow only!
My talk is about this arrow only!

- partial results are being processed immediately
- reaction is quicker, back-channels are possible

assess and improve information flow
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time (open video)
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time
- more edit than necessary → overhead ~ 90%!
  - 90% of a consumers work will be useless
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time
- more edit than necessary → overhead ~ 90%!
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time
- more edit than necessary \(\rightarrow\) overhead \(\sim 90\%\)
- reduce overhead, sacrifice some timeliness

which edits should we trust?

Patience, Young Jedi!
waiting helps
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time
- more edit than necessary $\rightarrow$ overhead $\sim 90\%$
- reduce overhead, sacrifice some timeliness

Software from Malsburg et al., submitted
Content: Basically we …

1. first say: „incremental behaviour is important!“
2. define measures to capture incremental behaviour
3. determine the incremental behaviour of our ASR
   - there are trade-offs between measures
4. develop ways to manipulate the behaviour
5. balance settings to suit our needs
Descriptive Measures for Incremental ASR

- there are three groups of measures
  - **accuracy**
  - **change**
  - **timing**
- measure against non-incremental ASR as our gold
  - we only measure incremental aspects, overall performance (WER/SER) is measured separately
- we focus on **words** only and ignore **silence markers** (<sil>)
A Reduced Example

- $w_{hyp_t}$ is the word sequence hypothesized at time $t$
- two dimensions:
  - time we reason about: $\rightarrow$
  - time we reason at: $\downarrow$
- $w_{gold}$ is final hypothesis
Accuracy Measures

Correctness of hypotheses

\textbf{r-correct:}

\[ w_{hyp_t} = w_{gold_t} \]

\textbf{p-correct:}

\[ w_{hyp_t} \text{ prefix-of } w_{gold_t} \]

(p-correctness adjusts for ASR lag at word boundaries)
Change Measure

- changes on the right
- *add*, *delete* or *revise*
- ideally: one *add* per word
- in fact: edit overhead

\[ EO = \frac{|\text{unnecessary edits}|}{|\text{edits}|} \]
Change Measure

ideally: 3 edits

actually: 11 edits

unwanted: 8 edits

EO: \( \frac{8}{11} = 72 \% \)
Edits are bad:

- edits lead to unnecessary processing of a consumer
  - less edits mean less processing

→ we would like to *reduce the edit overhead*
  → by *deferring* or *suppressing* edits

- deferring edits leads to delays, deteriorating *timing measures* …
Timing Measures

- when do we find out about a word?
  - word first correct: \textbf{WFC}
- when do we become certain about a word?
  - word first final: \textbf{WFF}
- this is per word
  - averages are important
Timing Measures

for "zwei":

first correct at $t = 7$

first final at $t = 9$

$\text{WFC}_{\text{zwei}} = 1$

$\text{WFF}_{\text{zwei}} = 0$

similarly for all other words
Timing Measures

• depending on the use-case we may care for …
  - if we want to assume as soon as possible → low WFC
  - if we want to know as soon as possible → low WFF

• deferring edits means two things:
  - higher WFC (as the lag passes through)
  - tendency for lower WFF (if we eliminate wrong edits)
Base Measurements

- **r-correct**: 30.9\%, **p-correct**: 53.1\%
- **edit overhead**: 90.5\%
  - most (9 of 10) edits are unnecessary!
- **WFC**: mean=0.276 s, stddev=0.186 s, median=0.230 s
  - average at \( \frac{3}{4} \) of the average word length
- **WFF**: mean=0.004 s, stddev=0.286 s, median=–0.06 s
  - final around word end (on average)

Sphinx-4 for German with statistical LM, WER = 18.8%, mean word length 0.378 s
Certainty Considerations

- the **correction time** for a word is **WFF–WFC**
- 58.6% of all words are immediately correct
- we can calculate the degree of **certainty** for given hypothesis ages
- e.g. if a correct hyp. lasts for 0.55 s, we can be certain (95%) that it will not change anymore
Improving Incremental ASR

- our primary goal is to reduce edit overhead
- ... by deferring or suppressing edits
  - deferring edits will always hurt WFC
  - suppressing edits may even improve WFF
  - the final (non-incremental) result does not change
    → only trust older parts of hyps. (Right Context)
    → only trust older edits (Message Smoothing)
Right Context to Improve Incremental Performance

- much jitter is at the right end of the hypotheses

  Æ at time $t$ only evaluate $hyp_t$ up to $t-\Delta$

- we need to take this into account for correctness:
  - fair r-correct: $w_{hyp_{t-\Delta}} = w_{gold_{t-\Delta}}$

- **WFC** increases with $\Delta$, **WFF** increases $\leq \Delta$

- we can predict the future with negative $\Delta$
  - e.g. fair r-correctness down 50% at 100 ms in the future
Message Smoothing to Improve Incremental Performance

- most bad edits only last for a short while
  - "zwei" → "zwar" → "zwei"

  → hold back edits until they reach a certain age
    - only output if they don't die before maturing

- multiple short edits of a word may delay messages:
  - **WFC** may grow without fixed bounds occasionally
  - probable resolution/mitigation: **future work** allow for some kind of "majority smoothing"
Right Context vs. Smoothing

EO parity (50%)

EO

Correctness

Fixed Lag

Smoothing

EO

delay in s (scale shows larger right contexts towards the left)
Right Context vs. Smoothing

Right Context:
- 530 ms bounded (≤530)
- EO parity (50%)
- Timing increase

Smoothing:
- 110 ms window low (+140/67 ms)
- Timing increase
Conclusion

- incremental behaviour is important!

- measures for incremental aspects of ASR
  - timing, overhead \(\rightarrow\) trade-offs between them

- methods to improve incremental aspects
  - analysis of the methods' characteristics on our ASR
  - combine? majority smoothing? \(\rightarrow\) future work

- determine operating point based on the analysis
  - e.g. overhead: \(9/10 \rightarrow \frac{1}{2}\), WFC/WFF: +140/67 ms
Thank You!

Acknowledgements:

Michaela Atterer and David Schlangen, my collaborators
DFG for funding (Emmy Noether programme)
Setup and Corpora

- Sphinx-4 (Walker et al., 2004), LexTree decoder, trigram LM
- KCoRS (IPDS, 1994) and OpenPento as training
- 85 semi-spontaneous utterances as test-set
- WER: 18.8%, SER: 68.2%
- average lengths of words: 0.378s, utterances: 5.5s

→ we disregard leading and trailing pauses in the evaluation of incremental performance
Variations of the Setup

- to test the stability of incremental measures, we
  - varied LM weights (to test LM influence) and
  - degraded audio quality (to test AM influence)
- WER changes radically with different LM weights (and especially with degraded audio)
- incremental measures (correctness, edit overhead) remain remarkably stable