Strongly Incremental Repair Detection

Julian Hough[1,2] and Matthew Purver[2]

[1]Dialogue Systems Group and CITEC, University of Bielefeld
[2]Cognitive Science Research Group, Queen Mary University of London

October 26th 2014, EMNLP Doha, Qatar

1. Problem statement
2. STIR: Strongly Incremental Repair Detection
   - Edit terms
   - Repair start
   - Reparandum start
   - Repair end
3. Evaluation measures for repair
4. Experiments and results
5. Conclusions and Future

1. Problem statement
2. STIR: Strongly Incremental Repair Detection
   - Edit terms
   - Repair start
   - Reparandum start
   - Repair end
3. Evaluation measures for repair
4. Experiments and results
5. Conclusions and Future

Self-repairs

"But one of **the, the** two things that I'm really..."

"Our situation is **just a little bit, kind of the** opposite of that"

"and you know it's like **you're, I mean,** employments are contractual by nature anyway"

[Switchboard examples]

Self-repairs: Annotation scheme

$$\text{John} \quad \underbrace{[\ \text{likes}}_{\text{reparandum}} \; + \; \underbrace{\{\text{uh}\}}_{\text{interregnum}} \; \underbrace{\text{loves}\ ]}_{\text{repair}} \text{Mary}$$

[Shriberg, 1994, onwards]

Terminology: *edit terms*, *interruption point* (+), *repair onset*

Self-repairs: classes

"But one of [the, + the] two things that I'm really…"

[repeat]

"Our situation is just [a little bit, + kind of the opposite] of that"

[substitution]

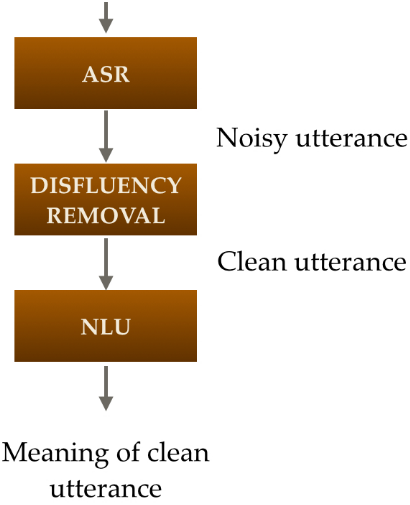
"and you know it's like [you're + {I mean}] employments are contractual by nature anyway"

[delete]

[Switchboard examples]

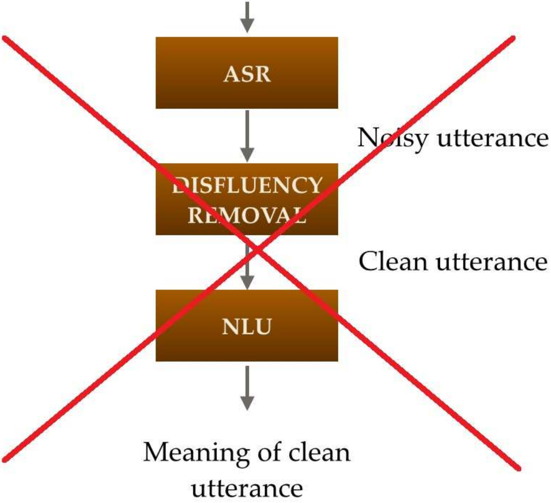
Self-repair detection: why do we care?

Dialogue systems (parsing speech)



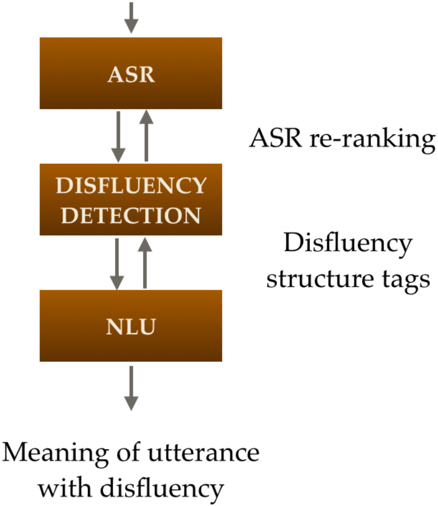
Self-repair detection: why do we care?

Dialogue systems (parsing speech)



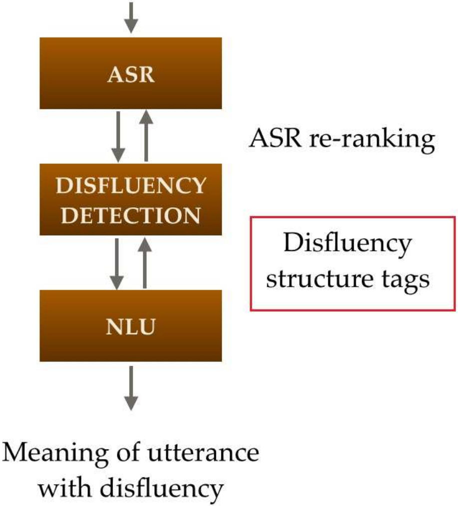
Self-repair detection: why do we care?

Dialogue systems (parsing speech)



Self-repair detection: why do we care?

Dialogue systems (parsing speech)



Self-repair detection: why do we care?

Interpreting self-repair

- Preserving the reparandum and repair structure
- Evidence: [Brennan and Schober, 2001] showed subjects use the reparandum to make faster decisions:

  "Pick the yell-purple square" *faster*

  "Pick the uhh-purple square"

Self-repair detection: why do we care?

Interpreting self-repair

- Preserving the reparandum and repair structure
- Evidence: [Brennan and Schober, 2001] showed subjects use the reparandum to make faster decisions:

  "Pick the yell-purple square" *faster*

  "Pick the uhh-purple square"
- Self-repairs have meaning!
- Dialogue systems should not filter out the reparandum!

Self-repair detection: why do we care?

Interpreting self-repair

- Preserving the reparandum and repair structure
- Evidence: [Brennan and Schober, 2001] showed subjects use the reparandum to make faster decisions:

  "Pick the yell-purple square" *faster*

  "Pick the uhh-purple square"
- Self-repairs have meaning!
- Dialogue systems should not filter out the reparandum!

Accuracy evaluation

- Standard evaluation F-score on reparandum words
- Also interested in repair structure assignment!

Self-repair detection: why do we care?

Interpreting self-repair

- Preserving the reparandum and repair structure
- Evidence: [Brennan and Schober, 2001] showed subjects use the reparandum to make faster decisions:

  "Pick the yell-purple square" *faster*

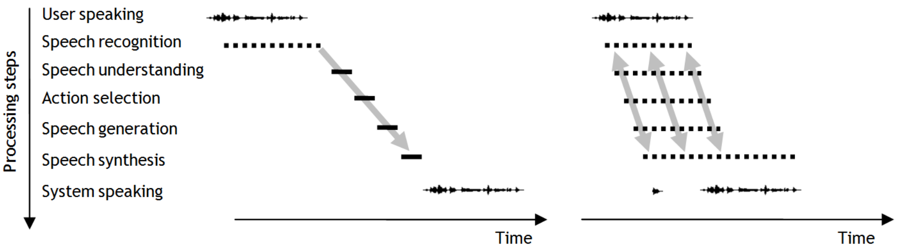
  "Pick the uhh-purple square"
- Self-repairs have meaning!
- Dialogue systems should not filter out the reparandum!

Accuracy evaluation

- Standard evaluation F-score on reparandum words
- Also interested in repair structure assignment!

Self-repair detection: Incrementality

Non-incremental vs. Incremental Dialogue Systems



[Schlangen and Skantze, 2011]

Self-repair detection: Incrementality

We want good *incremental performance*:

Timing

- Low latency, short time to detect repairs

Evolution over time

- Responsiveness of the detection (incremental accuracy)
- Stability of the output (low jitter)

Computational complexity

- Minimal processing overhead (fast)

Self-repair detection

Problem statement

A system that achieves:

- Interpretation of repair
  - *repair structure* tags rather than just reparandum words
- Strong incrementality
  - Give *the best results possible as early as possible*
  - Computationally fast
- Controllable *trade-off* between incrementality and overall accuracy

Previous approaches: Noisy channel model

- Best coverage generative model [Zwarts et al., 2010, Johnson and Charniak, 2004]
- S-TAG exploits (*'rough copy'*) dependency with string alignment
- [Zwarts et al., 2010] utterance-final F-score = 0.778

Previous approaches: Noisy channel model

- Best coverage generative model [Zwarts et al., 2010, Johnson and Charniak, 2004]
- S-TAG exploits (*'rough copy'*) dependency with string alignment
- [Zwarts et al., 2010] utterance-final F-score = 0.778
- Two incremental measures:
  - *Time-to-detection*: 7.5 words from reparandum onset
  - 4.6 words from repair onset
  - *Delayed accuracy*: slow rise up to 6 words back
- Complexity $O(n^5)$

Previous approaches: Noisy channel model

- Why poor incremental performance?

Previous approaches: Noisy channel model

- Why poor incremental performance?
- Inherently non-incremental string-alignment
- Utterance global (c.f. spelling correction)
- Sparsity of alignment forms [Hough and Purver, 2013]

SOLUTION: Information theory and strong incrementality

- *Local measures of fluency* for minimum latency in detection
- Does not just rely on string alignment
- Information theoretic measures of language models [Keller, 2004, Jaeger and Tily, 2011]
- Minimal complexity

1. Problem statement
2. STIR: Strongly Incremental Repair Detection
   - Edit terms
   - Repair start
   - Reparandum start
   - Repair end
3. Evaluation measures for repair
4. Experiments and results
5. Conclusions and Future

$$\text{John}\quad \underbrace{[\ \text{likes}}_{\text{reparandum}}\ +\ \underbrace{\{\text{uh}\}}_{\text{interregnum}}\ \underbrace{\text{loves}\ ]}_{\text{repair}}\ \text{Mary}$$

$$\ldots[rm_{start}\ldots rm_{end} + \{ed\}\ rp_{start}\ldots rp_{end}]\ldots$$

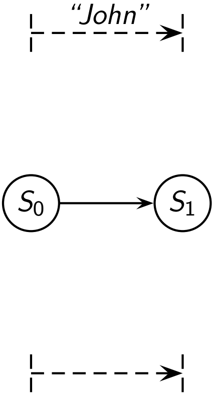
STIR: Strongly Incremental Repair Detection

$$...[rm_{start}...rm_{end} + \{ed\}\ \ rp_{start}...rp_{end}]...$$

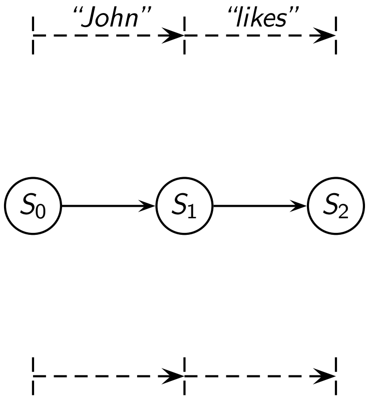
$$...\{ed\}...$$

STIR: Strongly Incremental Repair Detection

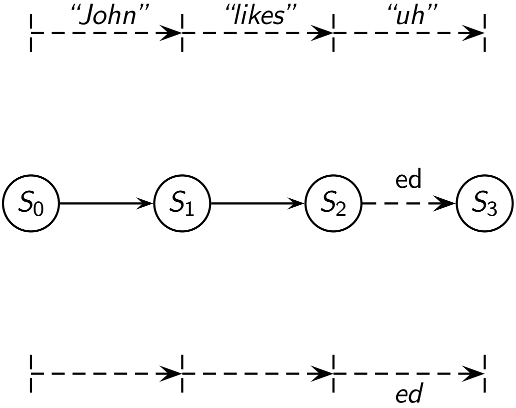
STIR: Strongly Incremental Repair Detection



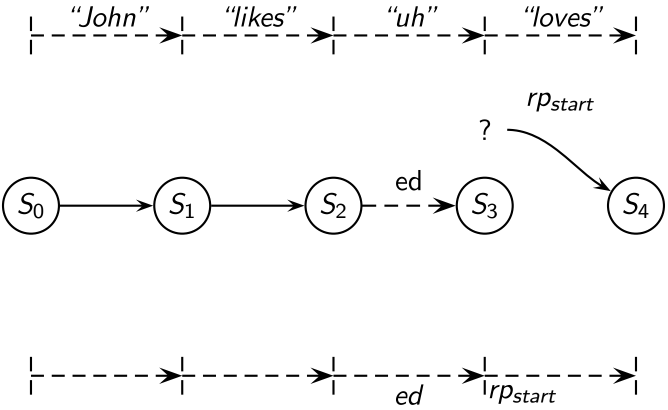
STIR: Strongly Incremental Repair Detection



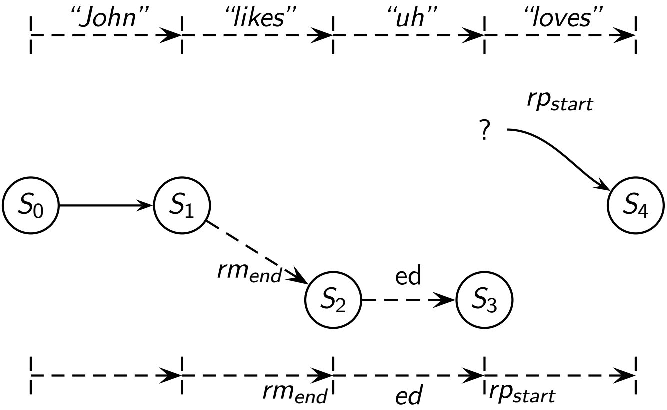
STIR: Strongly Incremental Repair Detection



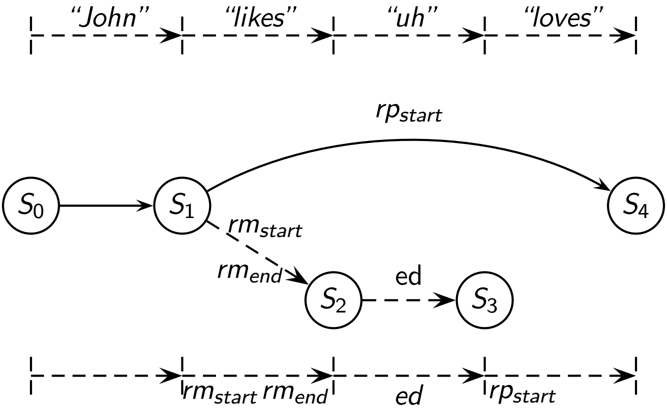
STIR: Strongly Incremental Repair Detection



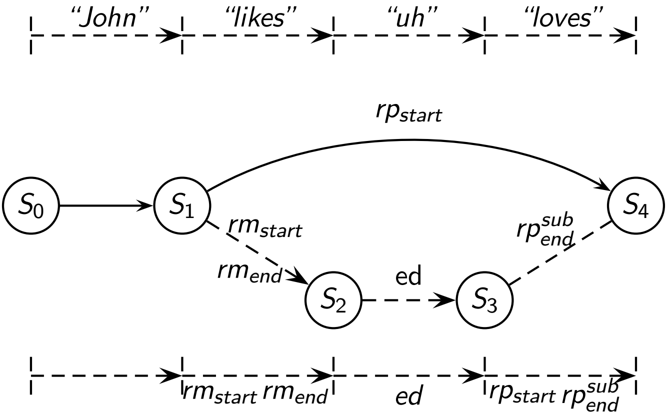
STIR: Strongly Incremental Repair Detection



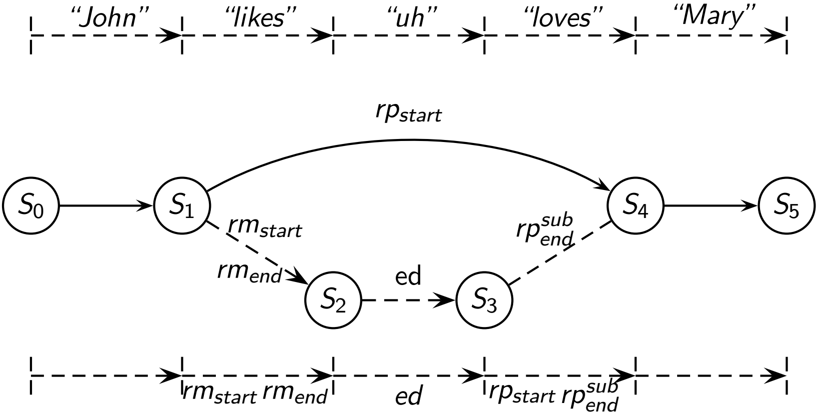
STIR: Strongly Incremental Repair Detection



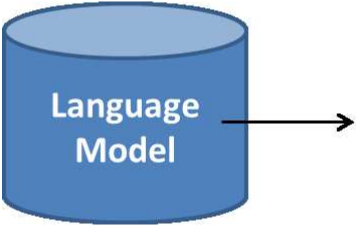
STIR: Strongly Incremental Repair Detection



STIR: Strongly Incremental Repair Detection



STIR: fluency modelling using enriched n-gram LMs



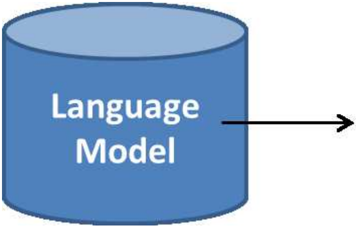
$s(w_{i-2}, w_{i-1}, w_i)$
(surprisal)

$\mathrm{WML}(w_{i-2}, w_{i-1}, w_i)$
(syntactic fluency)

$H(\theta(w \mid c))$
(entropy)

$KL(\theta(w \mid c_a), \theta(w \mid c_b))$
(distribution divergence)

STIR: fluency modelling using enriched n-gram LMs



$s(w_{i-2}, w_{i-1}, w_i)$
(surprisal)

$\mathrm{WML}(w_{i-2}, w_{i-1}, w_i)$
(syntactic fluency)

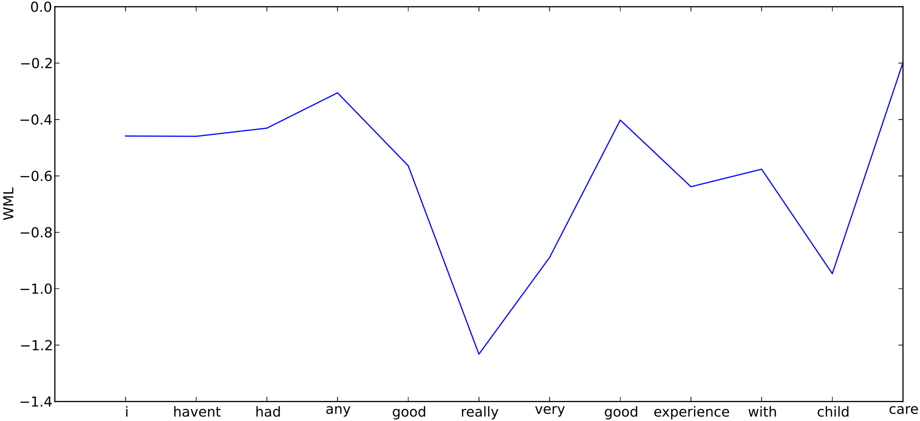
$H(\theta(w \mid c))$
(entropy)

$KL(\theta(w \mid c_a), \theta(w \mid c_b))$
(distribution divergence)

- p^{lex} (word) and p^{pos} (POS) models
- Does not use lexical or POS *values*, but information theoretic *measures* [Keller, 2004, Jaeger and Tily, 2011, Clark et al., 2013]

STIR: fluency modelling using enriched n-gram LMs

rp_{start} *local deviation from fluency*: drop in WML^{lex}



STIR: fluency modelling using enriched n-gram LMs

- Extend ‘rough copy’ dependency [Johnson and Charniak, 2004] to gradient measures
- Information content = *entropy*
- Parallelism = *distributional similarity*

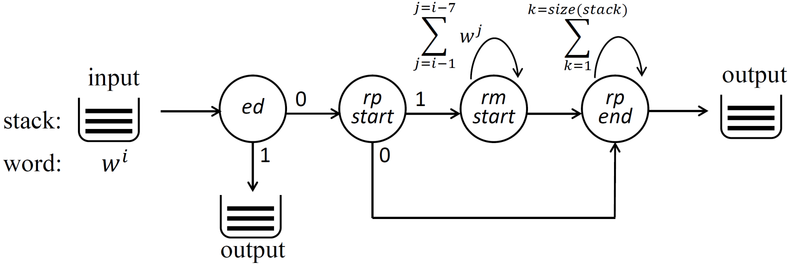
STIR: fluency modelling using enriched n-gram LMs

- Extend 'rough copy' dependency [Johnson and Charniak, 2004] to gradient measures
- Information content = *entropy*
- Parallelism = *distributional similarity*
- Repair-Reparandum correspondence = *gradient parallelism*

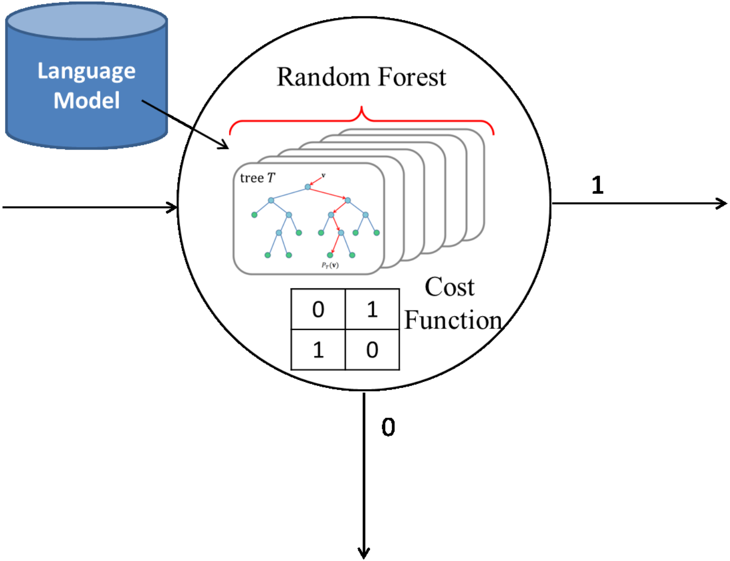
STIR: fluency modelling using enriched n-gram LMs

- **‘Fluent’ Language Model**: Trigram, Switchboard training data cleaned of disfluency (600K words)
- **‘Edit term’ Language Model**: Bigram, edit terms from Switchboard training data (40K words)

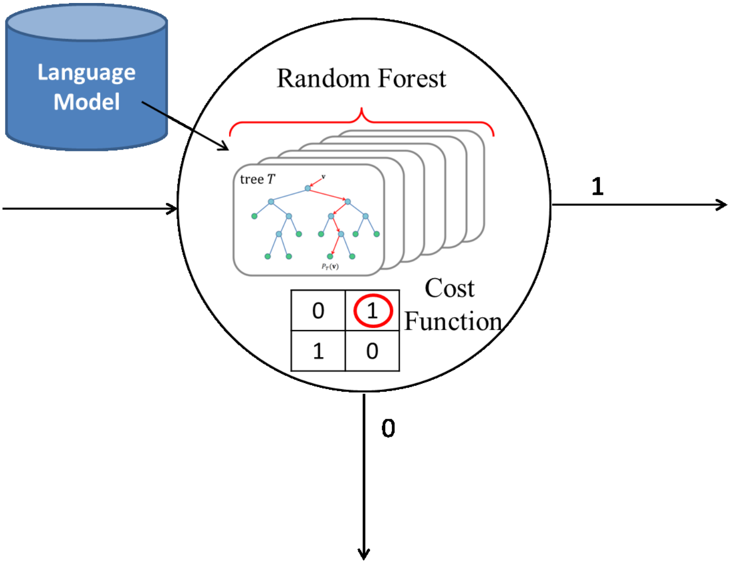
STIR: Classifiers



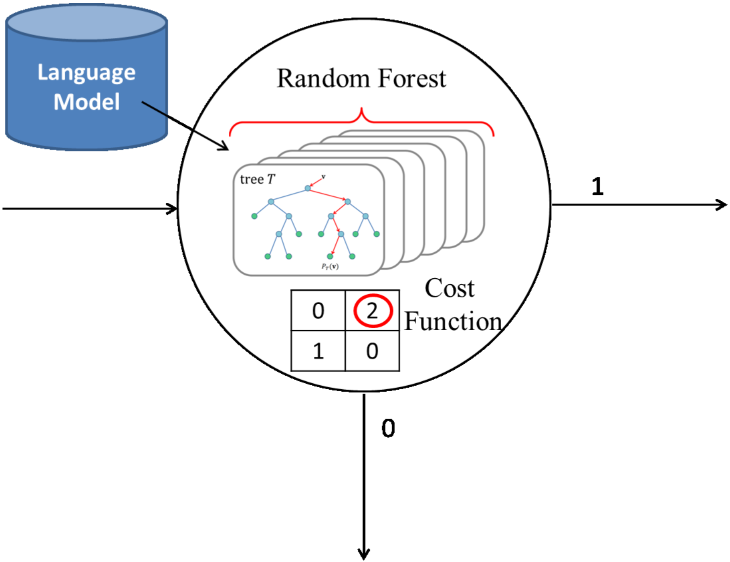
STIR: Classifiers



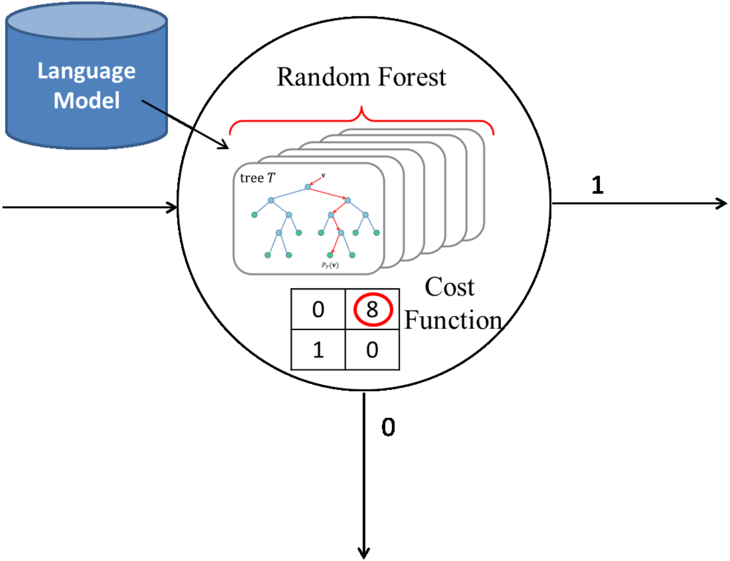
STIR: Classifiers



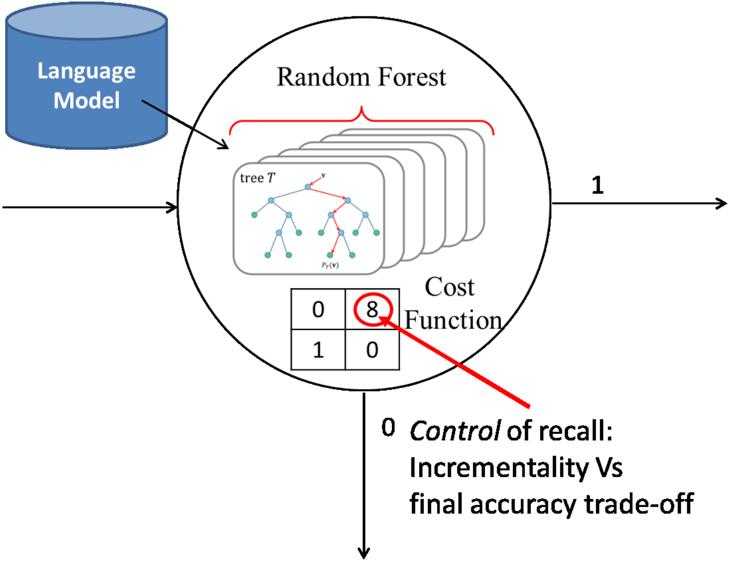
STIR: Classifiers



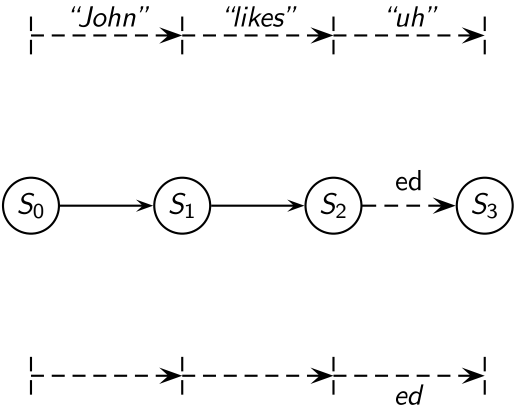
STIR: Classifiers



STIR: Classifiers



STIR: *ed* detection



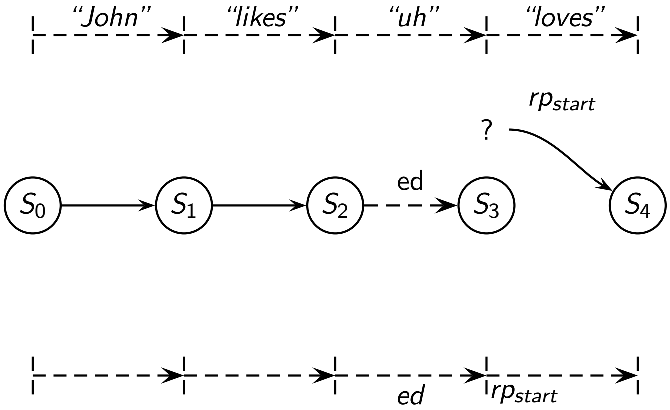
STIR: *ed* detection

- Edit term detection helps repair detection considerably
- Based on *WML* of words in edit term LM Vs. *WML* in fluent LM

STIR: *ed* detection

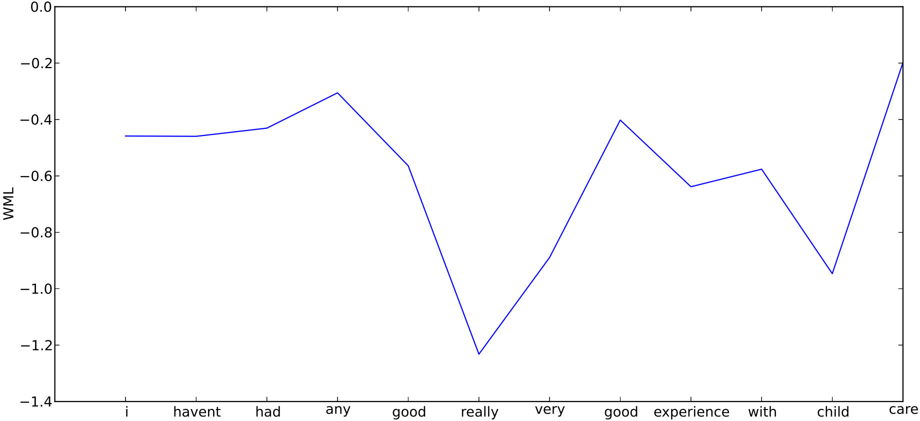
- Edit term detection helps repair detection considerably
- Based on *WML* of words in edit term LM Vs. *WML* in fluent LM
- Good performance: F-score **0.938** on *ed* words
- "I mean" and "you know" sometimes misclassified

STIR: rp_{start} detection



STIR: rp_{start} detection

rp_{start} *local deviation from fluency*: drop in WML^{lex}



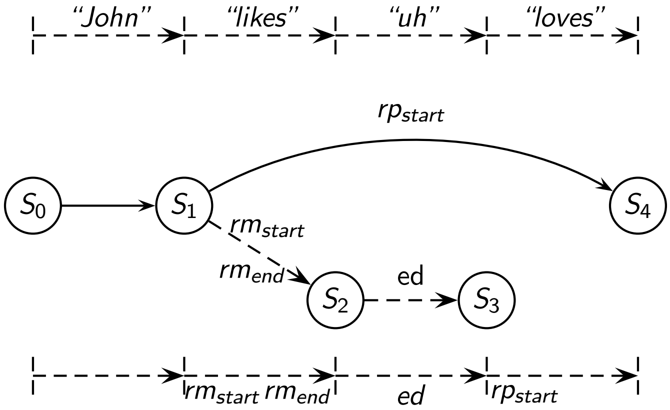
STIR: rp_{start} detection

- 23 features
- Best Features (ranking):

average merit	average rank	attribute
0.139	1	H^{pos}
0.131	2	WML^{pos}
0.126	3.4	WML^{lex}
0.125	4	s^{pos}
0.122	5.9	$w_{i-1} = w_i$
0.122	5.9	BestWMLBoost^{lex}

- LM features more useful than alignment in general
- Higher cost functions for false negs = higher recall

STIR: rm_{start} detection



STIR: rm_{start} detection

- 32 features
- Noisy channel intuition correct:
  - *WMLboost*:

    0.223 (sd=0.267) for rm_{start}

    -0.058 (sd=0.224) for other words in 6-word history
  - highest ranked feature is $\Delta WMLboost$
- Parallelism:
  - KL divergence between $\theta^{pos}(w \mid rm_{start}, rm_{start-1})$ and $\theta^{pos}(w \mid rp_{start}, rp_{start-1})$ second most useful feature

STIR: rm_{start} detection

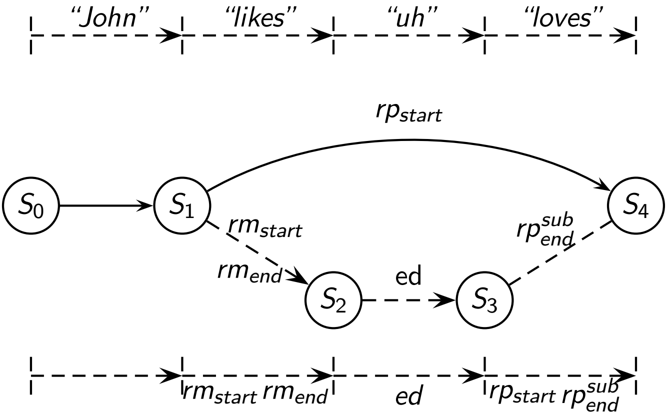
- Only allows backwards search to 7 words back
- Adds hypothesis to *stack* if rm_{start} found
- Complexity linear $O(n)$, in practice for most short utterances triangular $O(n^2)$

STIR: rm_{start} detection

- Only allows backwards search to 7 words back
- Adds hypothesis to *stack* if rm_{start} found
- Complexity linear $O(n)$, in practice for most short utterances triangular $O(n^2)$

- Control complexity increase with stack capacity:
- 1-best rm_{start} per rp_{start} $= O(n^2)$
- 2-best rm_{start} per rp_{start} $= O(n^3)$

STIR: rp_{end} detection



rp_{end} detection

- 23 features
- Parallelism:

- *ReparandumRepairDifference*: difference between *WML* of utterance with reparandum phase replacing repair and *WML* of utterance cleaned of reparandum

$$WML(\text{"John loves Mary"}) - WML(\text{"John likes Mary"})$$

- In both the POS and word model the best feature

rp_{end} detection

- 23 features
- Parallelism:

  - *ReparandumRepairDifference*: difference between *WML* of utterance with reparandum phase replacing repair and *WML* of utterance cleaned of reparandum

$$WML(\text{“John loves Mary”}) - WML(\text{“John likes Mary”})$$

  - In both the POS and word model the best feature
- Structural classification (repair extent)

1. Problem statement
2. STIR: Strongly Incremental Repair Detection
   - Edit terms
   - Repair start
   - Reparandum start
   - Repair end
3. Evaluation measures for repair
4. Experiments and results
5. Conclusions and Future

Evaluation

Accuracy

- Normal evaluation F-score on *rm* words (F_{rm})

Evaluation

Accuracy

- Normal evaluation F-score on *rm* words (F_{rm})
- Also interested in repair structure assignment (F_s)

Evaluation

Accuracy

- Normal evaluation F-score on *rm* words (F_{rm})
- Also interested in repair structure assignment (F_s)

Timing

- Time-to-detection rm_{start} and rp_{start} [Zwarts et al., 2010] (TD)

Evolution over time

- Delayed accuracy (of F_{rm}) [Zwarts et al., 2010] (DA)
- Edit overhead (stability) [Baumann et al., 2011] (EO)

Computational complexity

- Processing overhead (number of classifications per word) (PO)

Evaluation: Edit Overhead

Input and current repair labels					edits
John					
John *rm*	likes *rp*				$(\oplus rm)$ $(\oplus rp)$
John	likes	uh *ed*			$(\ominus rm)$ $(\ominus rp)$ $\oplus ed$
John	likes *rm*	uh *ed*	loves *rp*		$\oplus rm$ $\oplus rp$
John	likes *rm*	uh *ed*	loves *rp*	Mary	

- % of bad output edits
- *Repair gold standard* does not penalise *rm* before rp_{start}
- Therefore minimum (ideal) EO $= 0$

1. Problem statement
2. STIR: Strongly Incremental Repair Detection
   - Edit terms
   - Repair start
   - Reparandum start
   - Repair end
3. Evaluation measures for repair
4. Experiments and results
5. Conclusions and Future

Experiments

- **Training data (SWBD PTB)**: 650k words
- **Heldout data (SWBD PTB)**: 49K words
- **Test data (SWBD PTB)**: 48K words

Experiments

- **Cost functions**: 320 different settings used
- **Stack capacity**: 1-best rm_{start} and 2-best rm_{start} investigated

Results

Accuracy

- $\mathrm{F}_{rm} = \mathbf{0.779}$ for best setting
- Marginally improves [Zwarts et al., 2010]

Results

Accuracy

- $F_{rm} = \mathbf{0.779}$ for best setting
- Marginally improves [Zwarts et al., 2010]
- $F_s = \mathbf{0.736}$
- Novel metric. Repair structure assignment difficult for humans!

Results

Accuracy

- $F_{rm} = \mathbf{0.779}$ for best setting
- Marginally improves [Zwarts et al., 2010]
- $F_s = \mathbf{0.736}$
- Novel metric. Repair structure assignment difficult for humans!

Timing

- TD **1 word** from rp_{start}, **2.6 words** from rm_{start}, much improved

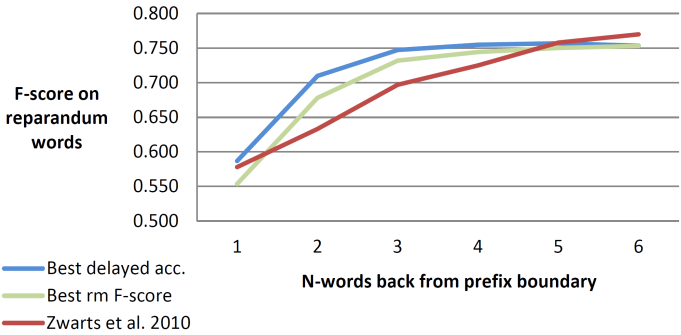
Results

Evolution over time

- EO varies, best very stable at **0.864**%

Evolution over time

- EO varies, best very stable at **0.864**%
- DA greatly improves:



Results

Computational complexity

- Limited to $O(n^2)$ and $O(n^3)$ in each stack setting a priori
- In practice very fast
- PO $=$ **1.229** per word in best setting

Results: trade-off

- In best final accuracy setting, high EO and PO (unstable and slower)
  - Requires high recall in rp_{start} classifier
- In most efficient and stable settings overall accuracy suffers

Results: trade-off

- In best final accuracy setting, high EO and PO (unstable and slower)
  - Requires high recall in rp_{start} classifier
- In most efficient and stable settings overall accuracy suffers
- Good trade-off setting found for incrementality and final accuracy
  - Fairly good $F_{rm} = \mathbf{0.754}$
  - Very low (good) EO $= \mathbf{0.931}$
  - Very low (good) PO $= \mathbf{1.255}$

1. Problem statement
2. STIR: Strongly Incremental Repair Detection
    - Edit terms
    - Repair start
    - Reparandum start
    - Repair end
3. Evaluation measures for repair
4. Experiments and results
5. Conclusions and Future

Conclusions

- STIR can experiment with final accuracy and incrementality trade-offs
- Achieves state-of-the-art latency and incremental performance in detection
- Detects entire repair structures - does not delete the reparandum!
- Does not use lexical or POS *values*, but information theoretic *measures*

Conclusions

- STIR can experiment with final accuracy and incrementality trade-offs
- Achieves state-of-the-art latency and incremental performance in detection
- Detects entire repair structures - does not delete the reparandum!
- Does not use lexical or POS *values*, but information theoretic *measures*
- STIR strongly incremental; useful for dialogue systems
- Currently being integrated with incremental ASR (DUEL project)

Thanks!

especially to:

- EPSRC DTA (Queen Mary University of London)
- DUEL project (Bielefeld University and Paris 7, DFG and ANR)

Baumann, T., Buß, O., and Schlangen, D. (2011).
Evaluation and optimisation of incremental processors.
Dialogue & Discourse, 2(1):113–141.

Brennan, S. and Schober, M. (2001).
How listeners compensate for disfluencies in spontaneous speech.
Journal of Memory and Language, 44(2):274–296.

Clark, A., Giorgolo, G., and Lappin, S. (2013).
Statistical representation of grammaticality judgements: the limits of n-gram models.
In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 28–36, Sofia, Bulgaria. Association for Computational Linguistics.

Domingos, P. (1999).
Metacost: A general method for making classifiers cost-sensitive.
In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164. ACM.

Honnibal, M. and Johnson, M. (2014).
Joint incremental disfluency detection and dependency parsing.
Transactions of the Association of Computational Linugistics (TACL), 2:131–142.

Hough, J. and Purver, M. (2013).
Modelling expectation in the self-repair processing of annotat-, um, listeners.
In *Proceedings of the 17th SemDial Workshop on the Semantics and Pragmatics of Dialogue (DialDam)*, pages 92–101, Amsterdam.

Jaeger, T. F. and Tily, H. (2011).
On language utility: Processing complexity and communicative efficiency.
Wiley Interdisciplinary Reviews: Cognitive Science, 2(3):323–335.

Johnson, M. and Charniak, E. (2004).

A TAG-based noisy channel model of speech repairs.

In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 33–39, Barcelona. Association for Computational Linguistics.

Keller, F. (2004).

The entropy rate principle as a predictor of processing effort: An evaluation against eye-tracking data.

In *EMNLP*, pages 317–324.

Qian, X. and Liu, Y. (2013).

Disfluency detection using multi-step stacked learning.

In *Proceedings of NAACL-HLT*, pages 820–825.

Rasooli, M. S. and Tetreault, J. (2014).

Non-monotonic parsing of fluent umm I mean disfluent sentences.

EACL 2014, pages 48–53.

Schlangen, D. and Skantze, G. (2011).

A general, abstract model of incremental dialogue processing.

Dialogue and Discourse, 2(1):83–111.

Shriberg, E. (1994).

Preliminaries to a Theory of Speech Disfluencies.

PhD thesis, University of California, Berkeley.

Zwarts, S., Johnson, M., and Dale, R. (2010).

Detecting speech repairs incrementally using a noisy channel approach.

In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1371–1378, Stroudsburg, PA, USA. Association for Computational Linguistics.

STIR: fluency modelling using enriched n-gram LMs

- Fluency: insights from grammaticality modelling [Clark et al., 2013]- Kneser-Ney smoothed trigram model

$$s(w_{i-2}, w_{i-1}, w_i) = -log_2 p_{kn}(w_i \mid w_{i-2}, w_{i-1})$$

- Approx. to *syntactic fluency*: Weighted Mean Logprob (WML) [Clark et al., 2013]

$$WML(w_i..w_n) = \frac{log_2 p_{kn}^{TRIGRAM}(\langle w_i..w_n \rangle)}{-log_2 p_{kn}^{UNIGRAM}(\langle w_{i+2}..w_n \rangle)}$$

STIR: fluency modelling using enriched n-gram LMs

- Subsume rough copy dependency [Johnson and Charniak, 2004] with gradient measures
- Quantifying uncertainty of continuing word through Shannon entropy:

$$H(w \mid c) = -\sum_{w \in Vocab} p_{kn}(w \mid c) \log_2 p_{kn}(w \mid c) \tag{1}$$

- Quantifying parallelism between reparandum and repair phases through KL divergence $KL(\theta(w_a \mid c_a), \theta(w_b \mid c_b))$
- Information content $=$ *entropy*
- Parallelism $=$ *distributional similarity*

- MetaCost error functions [Domingos, 1999] for false negatives
- Allows trade-off between incremental performance and final accuracy

$$\begin{array}{cc} & \begin{array}{cc} rp^{hyp}_{start} & F^{hyp} \end{array} \\ \begin{array}{c} rp^{gold}_{start} \\ F^{gold} \end{array} & \begin{pmatrix} 0 & 8 \\ 1 & 0 \end{pmatrix} \end{array}$$

Results: stack capacity

	F_{rm}	F_s	EO
1-best rm_{start}	**0.745**	**0.707**	3.780
2-best rm_{start}	**0.758**	**0.721**	4.319

Table : Comparison of performance of systems with different stack capacities