Modelling Expectation in the Self-Repair Processing of Annotat-, um, Listeners

Julian Hough and Matthew Purver

Cognitive Science Group
School of Electronic Engineering and Computer Science
Queen Mary University of London

Semdial, Amsterdam, 18th of December, 2013

Outline

Outline

Self-repairs

General form:

Self-repairs

General form:

Self-repairs

General form:

utterance

Self-repairs

General form:

utterance [*reparandum*

Self-repairs

General form:

utterance [*reparandum* +

Self-repairs

General form:

utterance [*reparandum* + {*interregnum*}

Self-repairs

General form:

utterance [*reparandum* + {*interregnum*}*repair*]

Self-repairs

General form:

utterance [*reparandum* + {*interregnum*}*repair*]*continuation*

Self-repairs

General form:

utterance [*reparandum* + {*interregnum*}*repair*]*continuation*
[Shriberg, 1994, onwards]

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

Self-repair classes

"But one of [the, + the] two things that I'm really..."
repeat *(sw4356)*

"Our situation is just [a little bit, + kind of the opposite] of that"
substitution *(sw4103)*

"and you know it's like [you're + {I mean}] employments are contractual by nature anyway"
delete *(sw4430)*

Self-repair classes: gradient judgements

"and [there's, + *?]* it's] completely generic."
substitution or delete? *(sw4619)*

"in the same token it's, [a very, + *?]* really] enjoyable for me because. . ."
substitution or delete? *(sw4109)*

"a matter where priorities are [at, +] placed.*?]*"
delete or substitution? *(sw4360)*

Self-repair classes: gradient judgements

"and [there's, + *?]* it's] completely generic."
substitution or delete? *(sw4619)*

"in the same token it's, [a very, + *?]* really] enjoyable for me because. . ."
substitution or delete? *(sw4109)*

"a matter where priorities are [at, +] placed.*?]*"
delete or substitution? *(sw4360)*

- Is categorical classification the right way to go?

Interpreting repair: Preserving reparanda

"[the interview was, + it was] alright"
substitution with anaphor in reparandum *[Clark, 1996]*

"Peter went [swimming with Susan, + {or rather,} surfing], on Tuesday."
substitution with trace *(Semdial reviewer)*

" [[I guess + I c-,] + I think] it's got some relevance, "
substitution- embedded *(sw4330)*

Interpreting repair: Incremental processing

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

  "Pick the uhh-purple square"

  "Pick the yell-purple square" *faster*
- Dialogue participants do not, and automatic systems should not, 'delete' the reparandum before downstream processing!
- Incremental interpretation taking place!

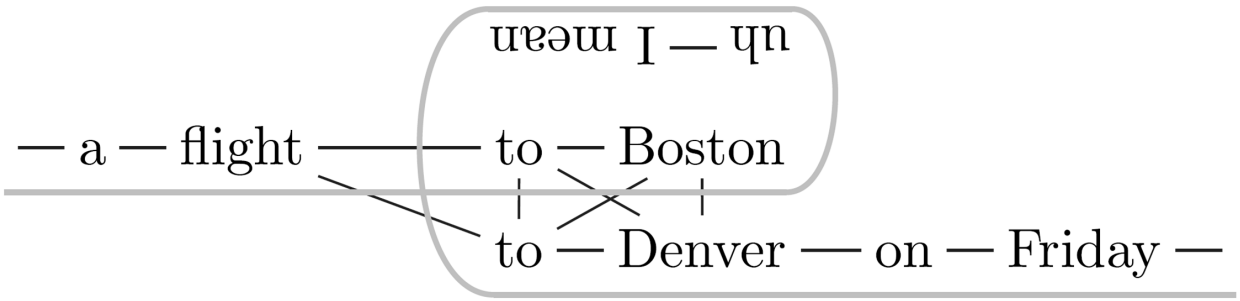
Previous work: Non-incremental

- [Qian and Liu, 2013] multi-stage learning with Max-Margin Markov networks. 3-stage.
  - (1) detection of edit-terms/fillers/interregna
  - (2) detection of reparandum words
  - (3) refining the previous steps, using a cost-sensitive error function.
  - F-score for reparandum words 0.841
- [Georgila, 2009] Integer Linear Programming (ILP).
  - Improves various off-the-shelf methods, best with CRF model.
  - f-score for detecting reparandum onset words = 0.808; repair onsets = 0.825.

Previous Work: Incremental

- Incremental detection: [Zwarts et al., 2010]'s word-by-word version of [Johnson and Charniak, 2004]
- S-TAG transducer [Johnson and Charniak, 2004] 'deletes' the reparandum, via generating:

  clean string $\rightarrow$ TAG rules $\rightarrow$ utterance (with repair).
- Their TAG repair model exploits crossing dependencies of the "rough copy":



- F-score 0.778 by end of utterances (same as non-incremental). Slight alteration to make early predictions.

Previous Work: Incremental

- Sparsity for language model (bigram). Incremental parser not available.
- Doesn't deal with embedded repairs.
- Complexity: TAG parsing $O(n^5)$.
- Chart of all possible repair structures grows cubically; [Heeman and Allen, 1999] suffer this too.

Previous Work: Incremental

- Sparsity for language model (bigram). Incremental parser not available.
- Doesn't deal with embedded repairs.
- Complexity: TAG parsing $O(n^5)$.
- Chart of all possible repair structures grows cubically; [Heeman and Allen, 1999] suffer this too.
- Incremental evaluation: latency (time-to-detection) and delayed accuracy scores (recall over all gold-tags in utterance prefixes).
  - Time-to-detection longer than average repair length: 7.5 words from reparandum start; 4.6 from the start of the repair
  - Delayed accuracy is low for one word back (0.578), rising steadily to 0.770 at 6 words back.

Criticism of systems

- Current systems are not inherently incremental, using over-prediction and filtering.
- Time-to-detection slow (latency).
- Computationally complex (psychologically unlikely).
- Not evaluated on *classifying* the self-repair processed, only for identifying reparandum words.

Criticism of systems

- Current systems are not inherently incremental, using over-prediction and filtering.
- Time-to-detection slow (latency).
- Computationally complex (psychologically unlikely).
- Not evaluated on *classifying* the self-repair processed, only for identifying reparandum words.
- Not really mechanisms for *repair*.

Approach


RONSEAL
DECKING
OIL
www.ronseal.co.uk
Does exactly what it says on the tin.

Approach

- A repair detector/classifier should REPAIR.

Time-linear process

Consider the order listeners process a self-repair:

(1) *Repair onset* detection, triggered via combination of a partial word, editing term forming an interregnum or fluency characteristics of the repair onset.

(2) *Reparandum start* position estimation through some backward-looking process.

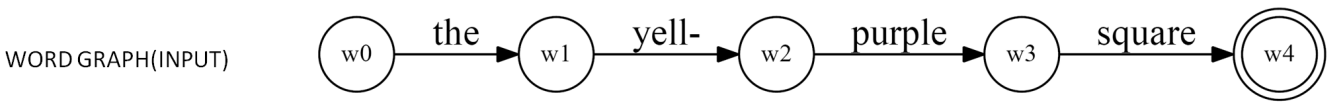
(3) Possibly interleaved with (2), estimation of the *repair end*, via detection of a further repair, fluent continuation or utterance end. Simultaneously with (1) and (2), *classification* of the repair.

Time-linear process: Formal model

- Parses word-by-word incrementally, compiles semantic formulae.
- If parse fails or no valid semantic formulae in domain: `REPAIR`:
  - Backtrack along processing context until successful parse and valid subsumption of a domain concept.

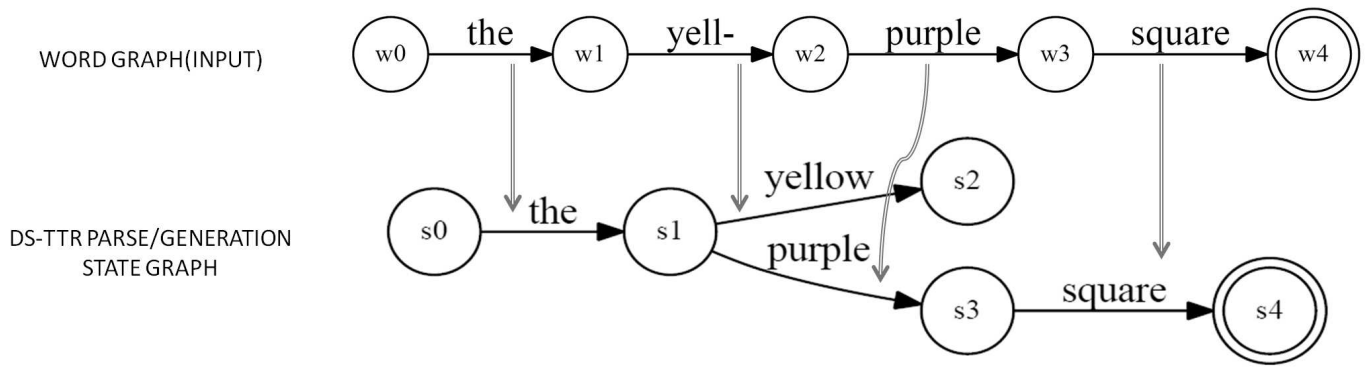
Time-linear process: Formal model

"The yell-purple square"



Time-linear process: Formal model

"The yell-purple square"



Time-linear process: Empirical questions

- Decision process for repair onset detection
  - Binary parseable vs. non-parseable $\{0,1\}$ or probablistic parsing [0,1]?
  - Edit term detection?
  - Partial words recognition?
- Decision process for reparandum onset detection
  - How far is necessary to backtrack? [Shriberg and Stolcke, 1998]
- Decision process for repair extent/classification
  - Parallelism vs. non-parallelism.

Outline

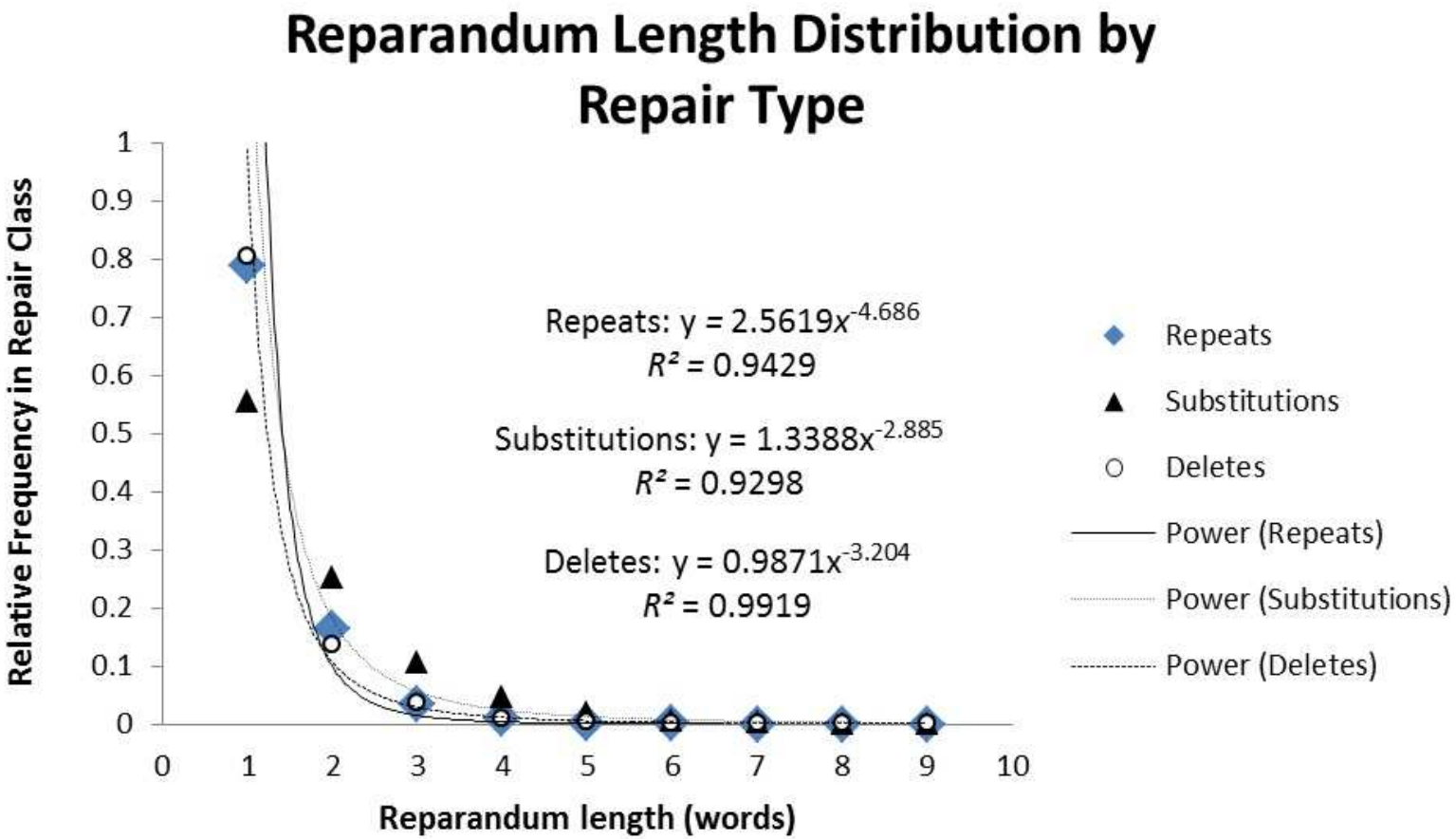
Switchboard

- From 972 transcripts, ∼196,600 utterances, ∼1.28M words, extracted 40,485 self-repairs based on the annotations.
- Base-rate likelihood of a repair onset = **0.0366**; once every 27.3 words of speech.
- Likelihood of a repair within an utterance unit = 0.1634.
- Likelihood of a second repair given one earlier in the utterance = **0.2563**.

Reparandum length

- Mean reparandum length 1.44 words (std.=0.88).
- Repeats (1.23 words) and deletes (1.35 words) are significantly shorter than substitutions (1.78 words).
- Inverse power law for reparandum lengths.
  - Reparanda of 1 or 2 words account for 90.8%
  - Lengths 1-3 account for 96.5%

Reparandum length

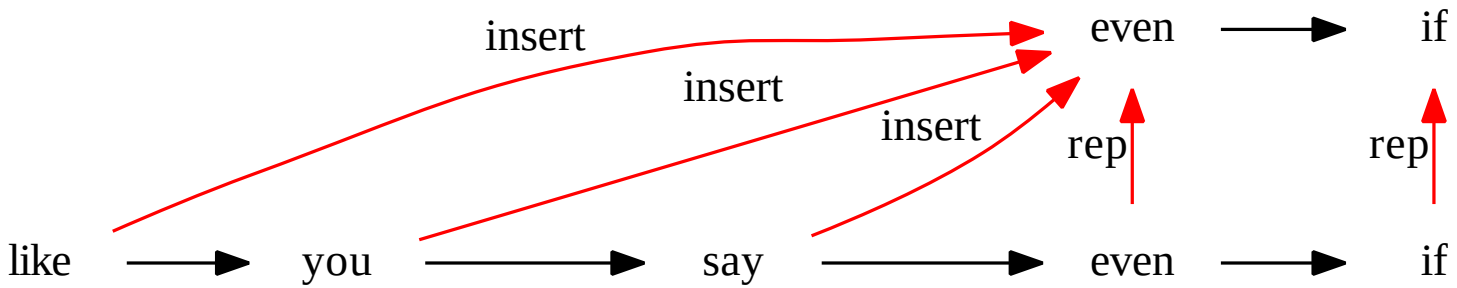


Class distributions

Class distributions

- Use weighted min. edit distance alignment [Johnson and Charniak, 2004]

. . . [even if, + like you say, even if] it's just the secretary (sw2959)



Class distributions

- 1139 different alignment sequence types found (cleaned of embedded repairs, 1479 uncleaned).
- Only 38.9% of types occur at least twice ([Heeman and Allen, 1999] 29.8%)
  - i.e. 61.1% only appeared once.

Top structures within classes (% overall repair tokens):

Repeats (56.8%)	Substitutions(36.6%)	Deletes(6.6%)
I rep I 46.2%	firm sub office 10.2%	and del when 5.0%

Class distributions: Repeats

- 56.79% of all repairs
- +interregnum=11.96% of class
- Reparandum length = 1.23 (std=0.53)
- Alignment sequence types = 7

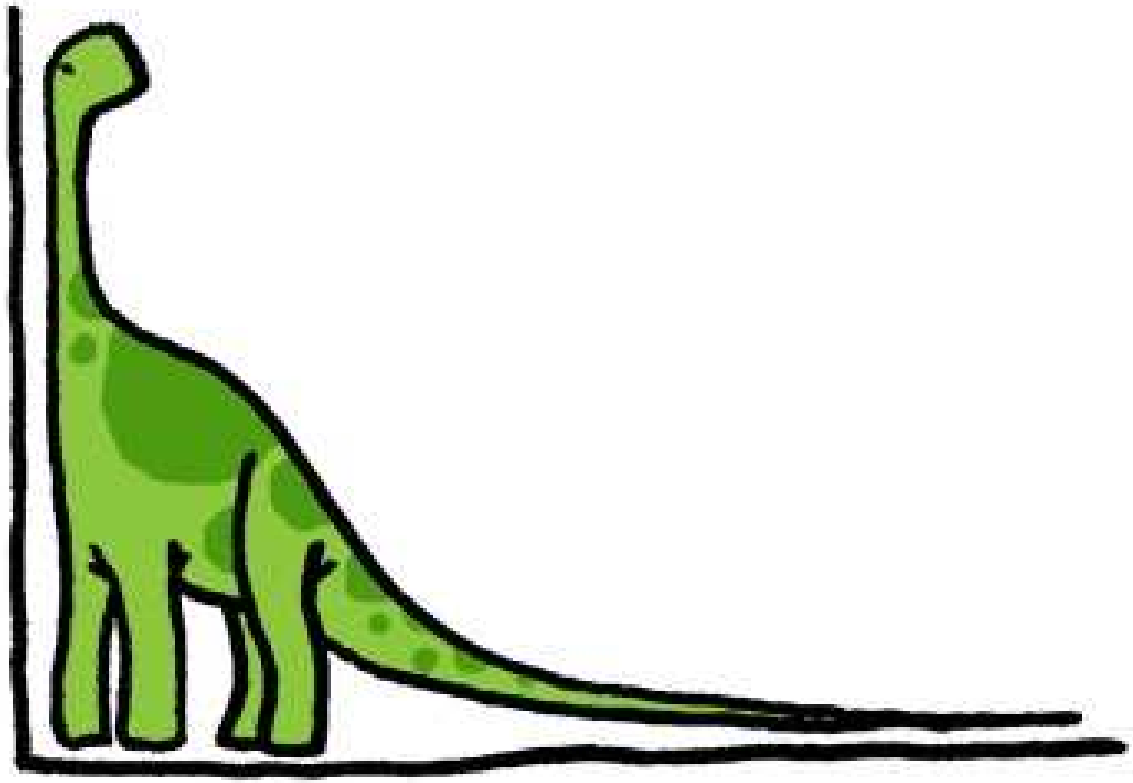
Class distributions: Substitutions

- 36.55% of all repairs
- +interregnum = 18.65% of class
- Reparandum length = 1.78 (std=1.16)
- Alignment sequence types = 1119

Class distributions: Deletes

- 6.66% of all repairs
- +interregnum = 0.7% of class
- Reparandum length =1.35 (std=0.88)
- Alignment sequence types = 13

Class distributions: Summary



Class distributions: Summary

- There is a long tail, mostly within substitutions.
- Is an alignment approach the right way to go?
  - For (short) repeats perhaps.
- Sparsity for training an aligner on substitutions.
- Rare repairs are long:
  - run-time of min. edit distance aligner $O(nm)$ [Jurafsky and Martin, 2009]
- S-TAG-like solution would reduce sparsity, but is slower $O(n^5)$
- Requires over-prediction over all possible repair sequences.

Predictive features: Embedded repairs

Predictive features: Embedded repairs

- " [[This, + it,] + they] are really. "
  Embedded chaining substitution- (sw3389)
- 11.9% of all repairs embedded within larger repair structure
- 9.9% chaining repairs; p(repair2onset|repair1end)=**0.110**
- 2.0% nested within repair phase;
  p(repair2onset|repair1onset)=0.020

Predictive features: Embedded repairs

- " [[This, + it,] + they] are really. "
  Embedded chaining substitution- (sw3389)
- 11.9% of all repairs embedded within larger repair structure
- 9.9% chaining repairs; p(repair2onset|repair1end)=**0.110**
- 2.0% nested within repair phase;
  p(repair2onset|repair1onset)=0.020
- Hierarchical structure or local linear indication of speaker trouble?

Predictive features: Interregna

- They are drawn from a characteristic vocabulary, and hence can be easily identified automatically [Heeman and Allen, 1999].
- Edit terms shift the class predictions marginally.

Predictive features: Interregna

- They are drawn from a characteristic vocabulary, and hence can be easily identified automatically [Heeman and Allen, 1999].
- Edit terms shift the class predictions marginally.
- However, they are not very predictive of repair. Need a better indicator.

Predictive features: Interregna

form	p(repair|form)	p(form|repair)
(fluent word)	0.037	0.861
"uh"	0.155	0.090
"you know"	0.100	0.026
"well"	0.080	0.006
"I mean"	0.074	0.005
"um"	0.061	0.003
"yeah"	0.038	0.002
"or"	0.017	0.002
"like"	0.014	0.003
"so"	0.005	0.001
"actually"	0.025	0.001

Predictive features: Partial words

- Most indicative feature of repair onset
- Likelihood of repair after partial word = **0.925** (not including utterance-final partial words)
- 10.9% of repairs have reparandum-final partial words
- 3.5% of repairs are $\mathit{prefix}(W) \rightarrow W$

Outline

Problem statement

$$\underbrace{\text{John and Bill}}_{\text{original utterance}} \quad \underbrace{[\ \text{like}}_{\text{reparandum}} \quad + \quad \underbrace{\{\text{uh}\}}_{\text{interregnum}} \quad \underbrace{\text{love}\ ]}_{\text{repair}} \quad \underbrace{\text{Mary}}_{\text{continuation}}$$

Problem statement

$$...w_o^N[w_{rm}^1...w_{rm}^N + w_{rp}^1...w_{rp}^N]w_c^1...$$

Problem statement

$$...w_o^N[w_{rm}^1...w_{rm}^N + w_{rp}^1...w_{rp}^N]w_c^1...$$

(0) (continuously) incrementally detect editing terms.

(1) incrementally detect w_{rp}^1

(2) IF (1) is True, backwards search for w_{rm}^1

(3) (interleaved with (2)), incrementally find w_{rp}^N and *classify*.

Fluency modelling: Adapted n-gram models

- Need a probabilistic model of fluency.
- Ideally incremental semantic parser (unavailable, in progress).
- Following work on grammaticality modelling [Clark et al.,2013], trigram model with Kneser-Ney smoothing.

$$p^{lex}(w_i \mid w_{i-2}, w_{i-1}) = p^{KN}(w_i \mid w_{i-2}, w_{i-1})$$

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

$$WML(w_i..w_n) = \frac{logp^{lex}_{TRIGRAM}(\langle w_i..w_n \rangle)}{logp^{lex}_{UNIGRAM}(\langle w_i..w_n \rangle)}$$

Fluency modelling: Adapted n-gram models

- Partial words key to repair detection as $p(w_i = w^1_{rp} \mid w_{i-1} = partial) = 0.925$
- Incorporate them into fluency measure.
- For a partial word w_i, the likelihood of w being its corresponding complete word at the time of interruption is:

$$p^{complete}(w \mid w_{i-2}, w_{i-1}, w_i) = \frac{1}{Z} \times p^{lex}(w \mid w_{i-2}, w_{i-1}) \times p^{prefix}(w \mid w_i)$$

where Z is a standard normalisation constant to ensure that $\sum_{w \in Vocab} p^{complete}(w \mid w_{i-2}, w_{i-1}, w_i) = 1$.

Fluency modelling: Adapted n-gram models

- The probability $p^{\hat{complete}}$ of most likely completion of w_i:

$$p^{\hat{complete}} = \max_w p^{complete}(w \mid w_{i-2}, w_{i-1}, w_i)$$

- Intuition is to find the most likely fluent word, given the partial word's context. "I remem-" more fluent/predictable that "S-".

Distributional Measures for classification: Continuations

- We want gradient, real valued rather than categorical ways of interpreting a repair.
- We want to measure surprise and parallelism to classify this.
- We consider the distribution of the next word in a context:
  - Distribution $\theta^{lex}(w \mid w_{i-1}, w_i)$
  - Entropy measure $H(\theta^{lex})$
  - Relative entropy (KL divergence) between two distributions of continuations.

Hypothesis 1. Detection

$$...w_o^N[w_{rm}^1...w_{rm}^N + w_{rp}^1...w_{rp}^N]w_c^1...$$

1. Repair onsets w_{rp}^1 with their 2-word context will have significantly lower mean p^{lex} values than non-repair transition trigrams (lower lexical-syntactic probability), and exhibit considerably bigger drops in *WML* (lower syntactic probability) than other fluent trigrams in the utterance so far.

Hypothesis 2. Reparandum start identification

$$...w_o^N[w_{rm}^1...w_{rm}^N + w_{rp}^1...w_{rp}^N]w_c^1...$$

2. Processing the utterance with the reparandum removed appropriately will significantly increase the *WML* of the utterance so far (similar intuition to the noisy channel approach), more so than other hypotheses for w_{rm}^1 given w_{rp}^1.

Hypothesis 3. (Initial) Classification

$$...w_o^N[w_{rm}^1...w_{rm}^N + w_{rp}^1...w_{rp}^N]w_c^1...$$

3. For repeats, the KL divergence from the continuation distribution after the reparandum's first word, i.e. $\theta^{lex}(w \mid w_o^N, w_{rm}^1)$, and that of the repair onset and its cleaned context before the reparandum, i.e. $\theta^{lex}(w \mid w_o^N, w_{rp}^1)$, will trivially be 0 in repeats and repeat-initiated substitutions, will be greater for other substitutions and higher still for deletes.

Hypothesis 4. Partial word repair classification

$$...w_o^N[w_{rm}^1...w_{rm}^N + w_{rp}^1...w_{rp}^N]w_c^1...$$

4. Repairs with reparandum-final partial words w_{rm}^N with high entropy over possible completions $\theta^{complete}$ or where the best completion has high KL divergence with the repair onset, will be interpreted as deletes rather than substitutions- in deletes the high uncertainty of predicted complete word is interpreted as 'cancelled'.

Hypothesis 5. Repair end detection/final classification

$$...w_o^N[w_{rm}^1...w_{rm}^N + w_{rp}^1...w_{rp}^N]w_c^1...$$

5. In repeats, the continuation distribution at the reparandum-final word w_{rm}^N (i.e. $\theta^{lex}(w \mid w_{rm}^{N-1}, w_{rm}^N)$) will be maximally close to that at the repair-final word w_{rp}^N (i.e. $\theta^{lex}(w \mid w_{rp}^{N-1}, w_{rp}^N)$) with KL divergence 0.

In substitutions, the same KL divergence will be on average higher than in repeats (though for compound type repairs ending in repeats this could still be 0), and the KL divergence for deletes should be even higher.

Hypothesis 5.i. Repair end detection/final classification

$$...w_o^N[w_{rm}^1...w_{rm}^N + w_{rp}^1...w_{rp}^N]w_c^1...$$

5.i. Substitutions as a class may vary significantly within this measure and in the KL divergence in hypothesis (3), however one KL divergence should be sufficiently lower than that of an average delete, and one should be higher than 0 due to them not being verbatim repeats.

Experiment 1. Low WML as repair onset indicator

Experiment 1. Low WML as repair onset indicator

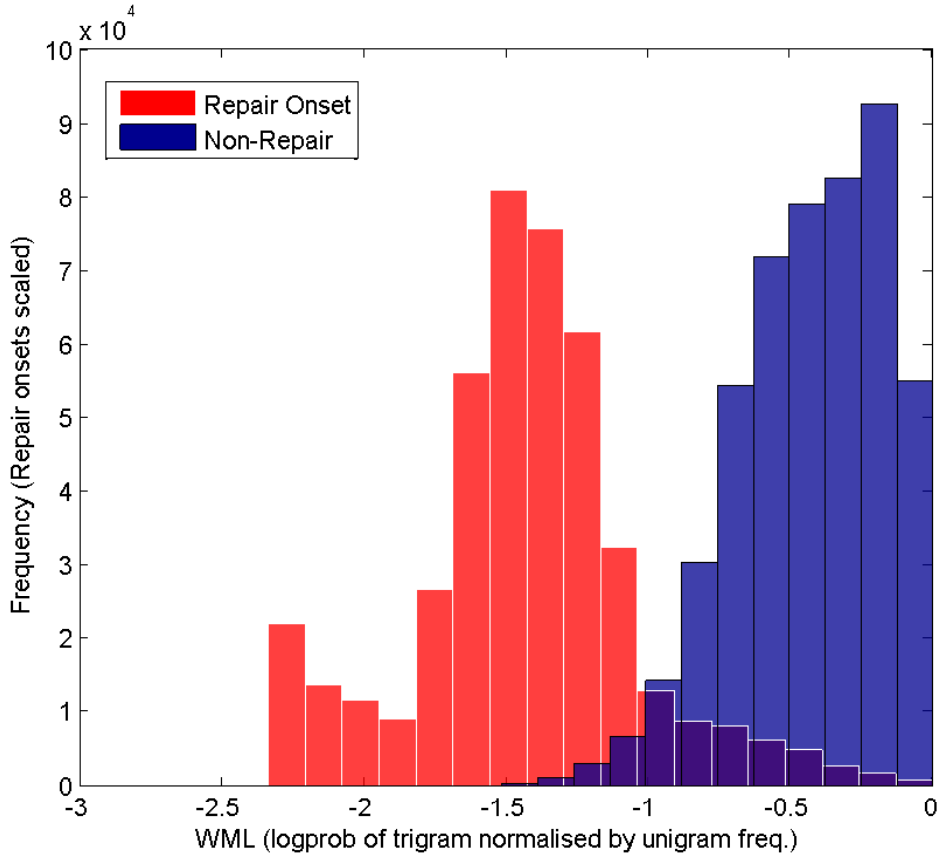
- Language Model: Switchboard training data (100K utterances, 650K words; fluent/cleaned- reparanda and edit-terms excised)
- Training data: Language Model corpus (with repairs/uncleaned)
- Unseen data: (Switchboard Heldout data, 6.4K utterances, 49K words)

Experiment 1. Low WML as repair onset indicator

- Language Model: Switchboard training data (100K utterances, 650K words; fluent/cleaned- reparanda and edit-terms excised)
- Training data: Language Model corpus (with repairs/uncleaned)
- Unseen data: (Switchboard Heldout data, 6.4K utterances, 49K words)

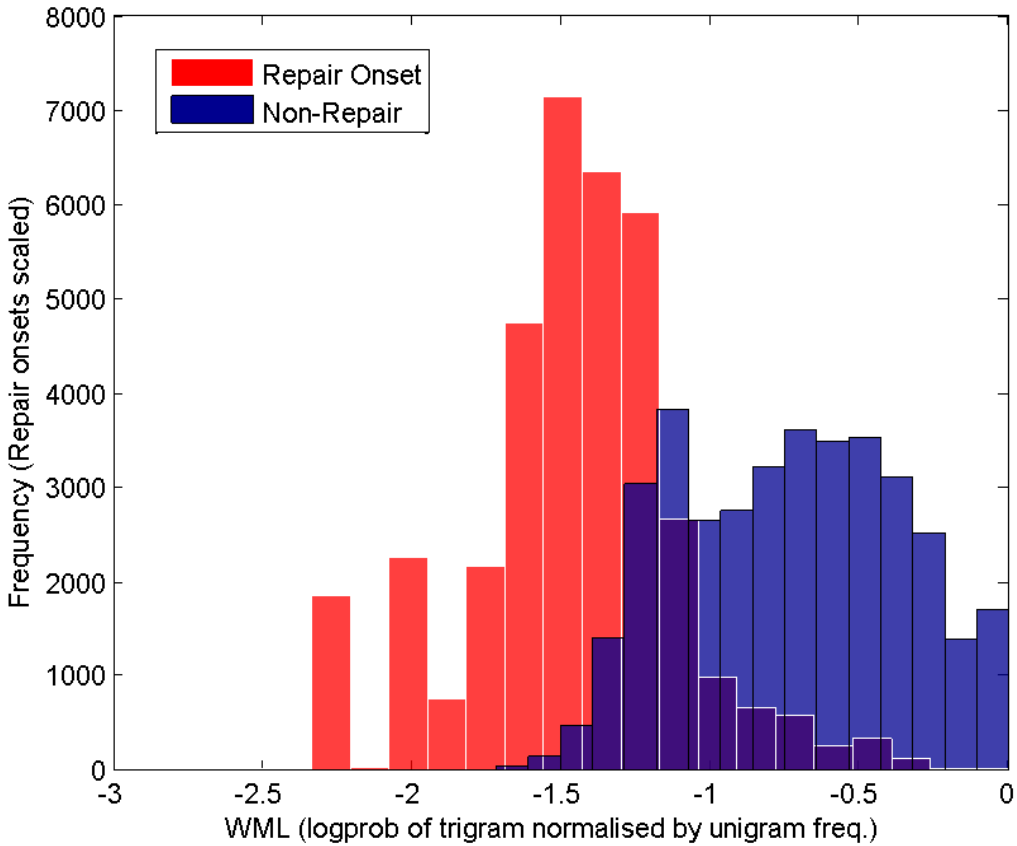
	Non Repair onsets	Repair onsets
Training data	-0.432 (std.=0.262)	-1.434 (std.=0.388)
Unseen data	-0.736 (sd=0.359)	-1.457 (std.=0.359)

Experiment 1. Low WML as repair onset indicator



Training data: WML fluency measure

Experiment 1. Low WML as repair onset indicator



Unseen data: WML fluency measure

Experiment 2. Utility of Local Fluency Features

- How useful are *WML* and p^{lex} compared to other local features for onset detection?

  Ngram features (3) : p^{lex}, *WML*, *WMLdrop*

  POS Ngram features (3) : $POSp^{lex}$, *POSWML*, *POSWMLdrop*

  Alignment (4) : *W2=W3*, *W1=W3*, *POS2=POS3*, *POS1=POS3*

  Edit terms (1) : *edit* [1,0]

Experiment 2. Utility of Local Fluency Features

- Feature ranker (Information Gain) 10-fold x-val, unseen data.

average merit	average rank	attribute
0.132 (+- 0.001)	1 (+- 0.00)	*WML*
0.123 (+- 0.004)	2.9 (+- 0.94)	$POSp^{lex}$
0.122 (+- 0.001)	3 (+- 0.63)	*W2=W3*
0.122 (+- 0.003)	3.1 (+- 0.83)	*POSWML*
0.084 (+- 0.001)	5.5 (+- 0.50)	*POS2=POS3*
0.086 (+- 0.003)	5.5 (+- 0.50)	*POSWMLdrop*
0.068 (+- 0.001)	7.3 (+- 0.46)	*WMLdrop*
0.066 (+- 0.003)	7.7 (+- 0.46)	p^{lex}
0.018 (+- 0.001)	9 (+- 0.00)	*W1=W3*
0.011 (+- 0)	10 (+- 0.00)	*edit*
0.008 (+- 0)	11 (+- 0.00)	*POS1=POS3*

Experiment 2. Utility of Local Fluency Features

- *WML* (syntactic) predictivity as measure of repair onset consistent with [Clark et al., 2013]. More predictive than alignment features.
- p^{lex}, raw probability alone perhaps suffers from sparseness.
- More repair examples may help alignment and edit features.

Experiment 3. Local Repair Onset Detection

- Simple classification of trigram's last word: w_{rp}^{1} or not.

Experiment 3. Local Repair Onset Detection

- Simple classification of trigram's last word: w_{rp}^{1} or not.
  - Strictly incremental detection (no latency or look-ahead)
  - Strictly local context used (trigram)

Experiment 3. Local Repair Onset Detection

- Simple classification of trigram's last word: w_{rp}^{1} or not.
  - Strictly incremental detection (no latency or look-ahead)
  - Strictly local context used (trigram)

- FIRST: Edit term detection, which helps considerably [Qian and Liu, 2013]
- Implemented incrementally- based on probability of word within an edit term vocabulary vs. its WML in fluent LM.
  - Very high accuracy (F-score 0.93+ over all edit term words)

Experiment 3. Local Repair Onset Detection

- Evaluation:

$$precision = \frac{w^1_{rp} correct}{w^1_{rp} hypothesised} \quad (1)$$

$$recall = \frac{w^1_{rp} correct}{w^1_{rp} gold} \quad (2)$$

$$Fscore = 2 \times \frac{precision \times recall}{precision + recall} \quad (3)$$

Experiment 3. Local Repair Onset Detection

Classifier: WEKA random forests (20 trees, 4 variables each)

(RESULTS: Non-partial words condition)

features	precision	recall	F-score
word ngram (3)	0.733	0.492	0.589
POS ngram (3)	0.722	0.518	0.604
All ngram (6)	0.829	0.561	0.669
Alignment (4)	0.942	0.546	0.691
All (11)	0.909	0.711	**0.798**

Discussion

- Repair onset detection comparable to [Georgila, 2009].
- Uses purely local features- no chart storage required and not complex.
- Latency minimal. Time-to-detection = 1 word from repair onset, greatly improves on [Zwarts et al., 2010]'s 4.6.

Discussion

- Repair onset detection comparable to [Georgila, 2009].
- Uses purely local features- no chart storage required and not complex.
- Latency minimal. Time-to-detection = 1 word from repair onset, greatly improves on [Zwarts et al., 2010]'s 4.6.
- Low recall due to lack of training data- experiments re-using data ongoing.
- Initial experiments with partial words condition ongoing- better score (0.83).
- Reparandum onset and repair extent experiments on-going.

Conclusions

- Current alignment techniques over-predictive and overly computationally complex. Not useful for long tail of types.
- Fluency features can predict repair onsets.
- Latency and computational complexity can be greatly reduced through useful *local* incremental features.
- Partial words a key part of language model for predicting presence and type of repair.
- Distributional parallelism/divergence can model gradient repair interpretations.

Thank you!

Thanks to Shalom Lappin and Gianluca Giorgolo among many others.

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.

Clark, H. H. (1996).

Using Language.
Cambridge University Press.

Georgila, K. (2009).

Using integer linear programming for detecting speech disfluencies.
In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 109–112. Association for Computational Linguistics.

Heeman, P. and Allen, J. (1999).

Speech repairs, intonational phrases, and discourse markers: modeling speakers' utterances in spoken dialogue.
Computational Linguistics, 25(4):527–571.

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*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.


Jurafsky, D. and Martin, J. (2009).
Speech and Language Processing.
Pearson Prentice Hall, 2nd edition.


Qian, X. and Liu, Y. (2013).
Disfluency detection using multi-step stacked learning.
In *Proceedings of NAACL-HLT*, pages 820–825.


Shriberg, E. (1994).
Preliminaries to a Theory of Speech Disfluencies.
PhD thesis, University of California, Berkeley.


Shriberg, E. and Stolcke, A. (1998).
How far do speakers back up in repairs? A quantitative model.
In *Proceedings of the International Conference on Spoken Language Processing*, pages 2183–2186.


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.