Computational Models of Miscommunication Phenomena

Matthew Purver

Cognitive Science Research Group,

School of Electronic Engineering and Computer Science,

Queen Mary University of London

Julian Hough

Dialogue Systems Group,

Faculty of Linguistics and Literature,

Bielefeld University

Christine Howes

Centre for Linguistic Theory and Studies in Probability (CLASP),

Department of Philosophy, Linguistics and Theory of Science,

University of Gothenburg

Keywords:

miscommunication, dialogue, repair, disfluency, incrementality, parallelism, sparsity

Corresponding Author: Matthew Purver, m.purver@qmul.ac.uk

School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom

Abstract

Miscommunication phenomena such as repair in dialogue are important indicators of the quality of communication. Automatic detection is therefore a key step towards tools which can characterize communication quality, and thus help in applications from call centre management to mental health monitoring. However, most existing computational linguistic approaches to these phenomena are unsuitable for general use in this way, and particularly for analysing human-human dialogue: although models of other-repair are common in human-computer dialogue systems, they tend to focus on specific phenomena (e.g. repair initiation by systems), missing the range of repair and repair initiation forms used by humans; and while self-repair models for speech recognition and understanding are advanced, they tend to focus on removal of "disfluent" material important for full understanding of the discourse contribution, and/or rely on domain-specific knowledge. We explain the requirements for more satisfactory models, including incrementality of processing and robustness to sparsity. We then describe models for self- and other-repair detection which meet these requirements (for the former, an adaptation of an existing repair model; for the latter, an adaptation of standard techniques), and investigate how they perform on datasets from a range of dialogue genres and domains, with promising results.

Computational Models of Miscommunication Phenomena

Repair Phenomena

One of the primary strategies by which interaction participants achieve and maintain shared understanding is *repair*: a set of strategies for highlighting and/or resolving instances of miscommunication or potential miscommunication. Not only are repair phenomena pervasive in conversation, and highly systematic, but their presence can reveal much about quality of communication, interaction and the participants themselves. A speaker can repair their own utterance, to adjust or clarify their talk (*"self-repair"*); this can be performed as the utterance is produced (example (1)), or later in a subsequent utterance (2). (In examples throughout, we show the *antecedent* (the material to be repaired) underlined, and the repair itself in **bold**.) These self-initiated examples reflect how hard speakers work on a turn-by-turn level to produce and fine-tune talk that is understandable to their specific conversational partner:

(1)[1]

Deb:	Kin you wait till we get home? We'll be home in five minutes.
Anne:	Ev//en less th'n that.
Naomi:	But c'd we– **c'd** I stay u:p?

(2)[2]

L:	I read a very interesting story today,
M:	uhm, what's that.
L:	**w'll not today, maybe yesterday, aw who knows when, huh**, it's called Dragon Stew.

However, a speaker can also repair another's utterance (3), or signal misunderstanding in order to elicit repair from the original speaker (4). These other-initiated examples (which we jointly term *other-repair* here) reflect how much effort speakers make to clarify understanding and address misunderstanding, in order to

[1](Schegloff, Jefferson, & Sacks, 1977) example (17)
[2](Schegloff et al., 1977) example (22)

reach shared understanding:

(3)[3]
Anon 3: Last year I was fifteen for the third time round.
Grace: Yeah. `<laugh>` Fifteen for the first time round.
Anon 3: **Third.**
Grace: Third time round.
Anon 3: Third time round.

(4)[4]
Sarah: Leon, Leon, sorry she's taken.
Leon: **Who?**
Sarah: Cath Long, she's spoken for.

Self-repairs are conventionally regarded as symptomatic of problems with communication on the part of the speaker, caused by self-monitoring or production issues (Bard, Lickley, & Aylett, 2001; Levelt, 1983). However, they are often associated with more interactive aspects of dialogue – many occur as we tailor our talk for specific addressees, or as a direct result of feedback from our interlocutors (Goodwin, 1979). There is also evidence that they do not just indicate miscommunication, but contribute to improving the effectiveness of interaction. For example, the presence of self-repairs can aid referential success (Brennan & Schober, 2001), affect grammaticality judgements (Ferreira, Lau, & Bailey, 2004) while leaving repaired material available for processing, and increase the frequency of backchannel responses by which listeners indicate their continued attention and understanding (Healey, Lavelle, Howes, Battersby, & McCabe, 2013). Other-repair too, despite the conventional view that it indicates negative aspects of miscommunication, has been shown to play a key role in semantic coordination (Mills & Healey, 2006), with evidence that increased levels of other-repair can improve task performance and speed up convergence on ways of referring (Mills, 2013).

Repair occurs across languages: cross-linguistic studies have shown that other initiation of repair is a standard function of questions, although the frequency of this can vary (see Stivers & Enfield, 2010, and others in that volume), and that many languages share the same repair mechanisms (Dingemanse et al., 2015) and even the surface form of the basic repair initiator "Huh?" (Dingemanse, Torreira, & Enfield, 2013). Rates of repair vary with a startling variety of factors, though; for example,

[3]BNC file KPE, sentences 326–331
[4]BNC file KPL, sentences 347–349

different domains and dialogue roles (Colman & Healey, 2011), modalities (Oviatt, 1995), dialogue moves (Lickley, 2001), gender and age groups (Bortfeld, Leon, Bloom, Schober, & Brennan, 2001). This is particularly well illustrated in the psychiatric domain, where aspects of doctor-patient communication are known to be associated with patient outcomes, in particular patient satisfaction, treatment adherence and health status (Ong, De Haes, Hoos, & Lammes, 1995), and studies specifically investigating repair show associations between repair and factors of clinical significance. Lake, Humphreys, and Cardy (2011) found that participants on the autistic spectrum revised their speech less often than controls, and used fewer filled pauses. For patients with schizophrenia, different rates of repair have been linked to assessments of the therapeutic relationship (McCabe, 2008), to specific types of symptoms such as verbal hallucinations (Leudar, Thomas, & Johnston, 1992), to difficulties with turn-taking that increase patient's social exclusion (Howes, Lavelle, Healey, Hough, & McCabe, 2017) and to likelihood of a patient's future adherence to their treatment (McCabe et al., 2013), with both self- and other-repair playing a significant role.

Detecting and characterising both self- and other-repair can therefore give useful information about the interactive processes which drive dialogue, the participants involved, and the quality of their interaction– see e.g. Healey, Colman, and Thirlwell (2005), who propose a repair-based approach to indexing coordination in dialogue. However, approaches of this kind have so far relied on time-consuming manual annotation. The ability to *automatically* quantify repair would therefore open a new opportunity for large scale (potentially real-time) dialogue analysis.

Existing computational models, however, have focused either on production mechanisms or on the needs of human-computer dialogue systems, rather than those of analysing human-human conversation. In the human-computer setting, the ability to deal smoothly with human input containing self-repair has long been recognised (see e.g. Johnson & Charniak, 2004), as has the need for systems to perform repair initiation when confidence in automatic speech recognition (ASR) or understanding is low (e.g. San-Segundo, Montero, Ferreiros, Córdoba, & Pardo, 2001); some recent systems can also perform self-repair in their own output when available information changes (Skantze & Hjalmarsson, 2010). However, current techniques are limited in their abilities (see next section); and recognition of repair initiation by users has received very little attention. As examples like (5) (from a real system prototyping session) show, though, the lack of this ability leads to errors if users try to initiate repair as usual:

(5)[5]

Sys:	Go East on Miranda Ave
Usr:	what is the next turn
Sys:	Make sharp right on Arastradero Rd
Usr:	**what road**
Sys:	I'm sorry, I do not understand what you mean by: what road. Please try rephrasing that.
Usr:	**what road do i turn on**
Sys:	I'm sorry, I do not understand what you mean by: what road do i turn on. Please try rephrasing that.
Usr:	**what road do i turn on** and where is alma street from there
Sys:	Did you want to plan a route from there?

The ability to automatically detect repairs is therefore important not only to provide tools for human-human interaction analysis (with potential applications including medical diagnosis and treatment monitoring), but also to improve human-computer dialogue systems as user behaviour becomes more natural. Here, we investigate models for self- and other-repair detection, and test how well they generalise between domains, with particular interest in the clinical domain.

Requirements and Existing Models

Types of Repair

In the conversation analysis (CA) literature (e.g. Schegloff et al., 1977), repair has long been a key subject of study, and is characterised in terms of who *initiates* the (need for) repair (oneself or another), who *completes* the repair (self or other), and in what *position* the repair is completed. Cases such as example (1) above, in which a speaker repairs their own utterance in the course of producing it, are thus termed *position one self-initiated self-repair* (P1SISR); repairing one's own antecedent utterance following an interlocutor's utterance, as in (2), a *position three self-initiated*

[5]Original data from prototype testing, CHAT project (Weng et al., 2007).

self-repair (P3SISR). An adjacent repair of another speaker's utterance, as in (3), is a *position two other-initiated other repair* (P2OIOR), and a clarification request as in (4) is a *position two next turn repair initiator* (P2NTRI). If the original speaker is then prompted to repair their problematic antecedent, as in the final utterance in each of (4), (6)–(9), this constitutes *position three other-initiated self repair* (P3OISR).

Colman and Healey (2011) show that by far the most common of these (more frequent than all other repair types combined), in both general conversation and task-oriented dialogue, is P1SISR self-repair (which is further subcategorised as articulation and reformulation), in line with CA's observations on the preference for self-repair in conversation (Schegloff et al., 1977). P2NTRI other-repair initiation is the next most common, and much more so than direct repair in that position (P2OIOR); responses to those in the form of P3OISR come next, with other types much less frequent. We therefore focus here on the most common forms of self- and other-repair (P1SISR, P2NTRI), noting also that McCabe et al. (2013) identify these as major informative factors in their predictive clinical model.

Even these categories, however, can take a variety of surface forms. P2NTRIs (or *clarification requests (CRs)*, see e.g. (Ginzburg & Cooper, 2004)) can appear not only as *wh*-words as in (4), but short fragments (6), longer reprises or echoes (but not necessarily verbatim) (7), and more explicit or conventional indicators (8)–(9) (Purver, Ginzburg, & Healey, 2003):

(6)[6]

Lara:	There's only <u>two people</u> in the class.
Matthew:	**Two people?**
Unknown:	For cookery, yeah.

(7)[7]

Anon 5:	<u>Oh he's started this other job</u>
Margaret:	**Oh he's started it?**
Anon 5:	Well, he he `<pause>` he works like the clappers he does!

[6] BNC file KPP, sentences 352–354

[7] BNC file KST, sentences 455–457

(8)[8]

Cassie:	You did get off with him?
Catherine:	Twice, but it was totally non-existent kissing so
Cassie:	**What do you mean?**
Catherine:	I was sort of falling asleep.

(9)[9]

Anon 2:	Gone to the cinema tonight or summat.
Kitty:	**Eh?**
Anon 2:	Gone to the cinema

Manual Analysis and Annotation

Healey et al. (2005) present a protocol for coding repair in interaction which identifies the different CA types of repair described above. Reliability of the protocol was shown to be encouraging — in an exercise re-coding a corpus of examples from the CA literature, 75% were assigned the same category as in the original — although detection agreement rates were not reported. Many more general annotation schemes for dialogue acts or utterance functions include repair initiation as a category (e.g. Jurafsky, Shriberg, & Biasca, 1997; Stivers & Enfield, 2010). Some use more fine-grained categorisations: P2NTRI repair initiators have been subcategorised according to various aspects of syntactic form, semantic structure and pragmatic level of intention (see e.g. Purver et al., 2003; Rodríguez & Schlangen, 2004). All such efforts we are aware of treat complete utterances or speaker turns as the candidate units for annotation: other-repair is by its nature a between-speaker phenomenon, and therefore naturally bounded by speaker changes.

Self-repair, on the other hand, can begin and end within a single speaker turn, so P1SISRs are often characterised using a word-level structural schema (Shriberg, 1994):

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

This structure affords three principal subtypes of self-repairs: *repetitions*, *substitutions* and *deletions*. Repetitions ('articulations' in CA terms) have identical reparandum and repair phases; substitutions have a repair phase that differs from its

[8] BNC file KP4, sentences 521–524

[9] BNC file KPK, sentences 580–582

reparandum phase lexically but is clearly substitutive of it; and deletions have no obvious repair phase that is substitutive of their reparandum, with utterance-initial deletions often termed *restarts* (both substitutions and deletions are 'reformulations' in CA). Despite the information such an approach provides, inter-annotator agreement is often low, and the consideration of gradient boundaries between categories may be more useful in some cases (Hough & Purver, 2013). Presence of a repair (or repair initiator) alone is agreed upon more often than structure or specific category.

Formal linguistic analyses of some repair mechanisms have been given, with some offering a unified treatment of self- and other-repair (e.g. Ginzburg, Fernández, & Schlangen, 2007); the differences in their form have so far kept annotation and computational approaches separate, though, and we maintain that distinction here.

Requirements for Models of Repair

These repair phenomena illustrate how dialogue participants manage and resolve (potential) misunderstandings as they arise, through and within interaction. For any computational model that hopes to capture them, whether in order to analyse human-human conversation, or produce a human-like dialogue system, this imposes several fairly challenging requirements; and few existing computational models meet these requirements with any degree of generality.

Parallelism with context. While both self- and other-repair can take many forms (1)–(9), all involve a reference to the antecedent material in context; ascribing a semantic interpretation must therefore require a model of this context (see e.g. Purver et al., 2003). Even if detection, rather than full interpretation, is the focus, many forms (e.g. the very common reprise NTRI forms in (4), (6)) can only be interpreted by detecting this reference via some form of similarity or parallelism with the antecedent; while many self-repair models are based on this, most other-repair models are not. This must go beyond simple lexical or syntactic repetition: some cases exploit similarities which are semantic (11), phonological (12) or even orthographic (13), and might be understood by one participant but not intended by the other (13):

(11)[10]
Dr: Are you suspicious are you suspicious of people
P: Suspicious
Dr: Paranoid
P: **Jealous**
Dr: Jealous yeah

(12)[11]
Dr: Paroxitine
P: **Fluoxitine**
Dr: Ah Fluoxitine

(13)[12]
Usr: how long
Wiz: dave's house is six minutes away
Usr: **was that one six or six zero minutes**
Wiz: six minutes away

Incrementality. Repair phenomena are inherently incremental: both self- and other-repair often occur mid-utterance with little regard for conventional notions of grammatical constituency or completeness (Howes, Purver, Healey, Mills, & Gregoromichelaki, 2011) – see (14). Detection models must be able to operate over incomplete utterances; in the case of human-computer dialogue systems, reacting suitably as soon as is appropriate.

(14)[13]
A: And er they X-rayed me, and took a urine sample, took a blood sample. Er, the doctor
B: **Chorlton?**
A: Chorlton, mhm, he examined me, erm, he, he said now they were on about a slide `<unclear>` on my heart.

A model for other-repair detection can rely on speaker changes to indicate potential repair points, but must be able to handle incomplete context and antecedent material. A self-repair detection model, however, must operate incrementally at a finer-grained level, considering individual words and even partial words.

Monotonicity. Another key requirement that stems from the incrementality of language processing is that the reparandum must be kept available for future processing. Psycholinguistic evidence shows that people do not discard repaired material (Brennan

[10] Doctor-patient interaction data, (McCabe et al., 2013).
[11] Doctor-patient interaction data, (McCabe et al., 2013).
[12] Original data from prototype Wizard-of-Oz testing, CHAT project (Weng et al., 2007).
[13] BNC file KPY, sentences 1005–1008

& Schober, 2001; Ferreira et al., 2004), and a model of context cannot therefore remove or overwrite antecedents, which can be anaphorically referred to (15), or crucial in the final interpretation of the utterance (16) (see Hough & Purver, 2012).

(15)[14] | Nancy: Um The interview was, **it** was alright

(16)[15] | A: Peter went swimming with Susan, or rather **surfing**, yesterday

Robustness to sparsity. Repair phenomena can be sparse. This is particularly clear for other-repair: P2NTRIs typically make up only 3-6% of utterances (3-4% (Purver et al., 2003), 5.8% (Rodríguez & Schlangen, 2004), 5.1% (Rieser & Moore, 2005)). However, in some domains, rates can be much lower: in the clinical dialogue domain of interest here, rates of P2NTRIs in patient speech can be as low as 0.8% (McCabe et al., 2013). Self-repair is, on the face of it, much more common, with 16-24% of utterances in general conversation containing a P1SISR (Hough, 2015); however, the proportion of *words* which begin a P1SISR is low (3.7-5.3%, Hough, 2015; Hough & Purver, 2013). As P1SISR is a within-utterance phenomenon, in which any word could potentially begin a repair, the sparsity problem is therefore still very real.

Computational Models

Despite progress in psycholinguistic modelling of production problems, most notably by Levelt (1983, 1989), most practical computational self-repair models have been designed for use in ASR and dialogue systems; while detection accuracy can be high, most take an approach of 'cleaning' speech of disfluent elements. This means they generally remove reparanda (antecedents), operate non-incrementally, and rely on relatively domain-specific dependency parsing rather than more general parallelism (e.g. Honnibal & Johnson, 2014; Rasooli & Tetreault, 2014) – thus failing to meet our requirements above. Some recent systems are incremental, and use more general statistical language model information (Zwarts, Johnson, & Dale, 2010), but still focus on removing antecedent material, not meeting our monotonicity requirement. They also

[14]From H. H. Clark (1996, p266)
[15]From Hough and Purver (2012)

generally use cleaned-up data with cut-off words removed. In contrast, the model we use below (STIR, Hough & Purver, 2014) meets all our incremental, domain-general, context-maintaining requirements and here we adapt it to handle cut-off words.

Computational models of other-repair initiation have generally focused on production, allowing systems to clarify errorful ASR input. Naturalness is typically limited (see (17), from the Let's Go! system, Raux, Langner, Black, & Eskenazi, 2005), although recent developments permit more natural, targeted NTRIs where uncertainty can be localised (18) (Stoyanchev, Liu, & Hirschberg, 2014):

(17)[16]
U: When's the next bus to Wood Street?
S: **Sorry, I didn't understand that. Please repeat.**
...
U: When's the next bus to Wood Street?
S: **Going to WOOD STREET. Did I get that right?**
U: Yes.

(18)[17]
U: Do you have anything other than these [XXX] plans
S: **Which plans? / Anything other than what?**

On the interpretation side, attention has been given to user correction (see e.g. Kitaoka, Kakutani, & Nakagawa, 2005; Lemon & Gruenstein, 2004; Litman, Hirschberg, & Swerts, 2006). When users notice system errors, they produce P2OIOR repairs, often using characteristic syntactic and prosodic forms (e.g. repetition with hyperarticulation) which then cause further misrecognition problems. Detection of corrections can therefore aid error recovery, and accuracies can be good (Kitaoka et al. (2005) report c.90% F-scores, although Litman et al. (2006) only 72% on different data, and Lopes et al. (2015) similar levels on a specific sub-task, repetition detection).

Recent approaches use general learning frameworks to induce these functionalities from data (see e.g. Young, Gašić, Thomson, & Williams, 2013), but do this by learning the optimal action for systems to take in a given context; this does not therefore directly generalise to detecting clarification by human users. While strategies for responding to user NTRIs could certainly be learned in principle, we are not aware of

[16]Let's Go! system examples (Stoyanchev & Stent, 2012).

[17]From (Stoyanchev et al., 2014); [XXX] represents a missing or unrecognised word.

current implementations; and these would not be suited to third-party analysis, being dependent on system interaction in the dialogue.

Dialogue act tagging tools, on the other hand, are designed for third-party analysis; however, they tend to be optimised for general overall accuracy, leading to relatively poor results for sparser classes, including repair and repair initiation. Much work does not attempt to classify these sparse classes (e.g. Stolcke et al., 2000); where results are given, accuracies are poor. Surendran and Levow (2006) report 43% F-scores on their P2NTRI category (`check`, 8% of turns in their dataset) and only 19% for P3OISRs (`clarify`, 4% of turns); Schlangen (2005) reports 30-40% F-scores on similar classes. Fernández, Ginzburg, and Lappin (2007) report good accuracies but only for a restricted sub-type of P2NTRIs (elliptical noun phrase fragments).

Below, we outline and test our own approach to general detection of repair and repair initiation, suitable for human-human as well as human-computer data and compatible with the requirements outlined above.

Materials

Corpora

Switchboard (SWBD). Our first corpus is one commonly used for testing computational self-repair models and and dialogue act taggers. The Switchboard corpus (Godfrey, Holliman, & McDaniel, 1992) consists of approximately 2400 dyadic telephone conversations between American participants unfamiliar with one another, on topics assigned from a pre-determined list. For other-repair, we use the Dialogue Act version of Switchboard (Jurafsky et al., 1997), with 1155 dialogues totalling over 120,000 utterances and nearly 1.5m words. For self-repair, we use the disfluency-tagged portion of Switchboard (Meteer, Taylor, MacIntyre, & Iyer, 1995), with 650 conversations of duration 1.5-10 minutes (average around 6.5 minutes), with a standard division into train, heldout and test sections (see Hough & Purver, 2014; Johnson & Charniak, 2004).

British National Corpus (BNC-CH, BNC-PGH). We also investigate how well our methods generalise to more open-domain and multi-party conversation. The

BNC-CH corpus (Colman & Healey, 2011) is a subset of the demographic portion (transcribed spontaneous natural conversations made by members of the public) of the British National Corpus (BNC, Burnard, 2000). It contains 31 dialogues annotated for self- and other-repair, with 1933 utterances, 14,034 words produced by 41 people. The BNC-PGH corpus (Purver et al., 2003) is a different, larger subset (c.150,000 words) containing sections from 56 dialogues including specific contexts (e.g. doctor-patient conversations) as well as demographic data, and annotated only for other-repair initiation (in their terminology, clarification requests).

Psychiatric Consultation Corpus (PCC). To test applicability to a clinical domain, we use a corpus from a study investigating clinical encounters in psychosis (McCabe et al., 2013): transcripts from 51 outpatient consultations of patients with schizophrenia and their psychiatrist, including 51 different patients, and 17 psychiatrists. Consultation length varies from only 709 words (c.5 minutes) to 8526 (nearly an hour), with mean length 3500 words.

Map Task Corpus (MAPTASK). To further investigate robustness to change in dialogue style, genre and domain, we also use the HCRC Map Task Corpus (Anderson et al., 1991), with 128 two-person dialogues containing 18,964 turns with c.150,000 words. These conversations concern a very specific task: guiding an interlocutor around a map whose features may not appear identical to the two parties.

Annotation

SWBD's disfluency annotations include filled pauses, discourse markers, and edit terms, all with standardised spelling (e.g. consistent 'uh' and 'uh-huh' orthography). P1SISRs are bracketed with the structure in (10), with reparandum, interregnum and repair phases marked. Restart repairs (utterance-initial deletions) are coded as two separate units and not in fact annotated as repairs. In the dialogue act corpus, P2NTRIs are tagged as *signal-non-understanding* (`br`); Jurafsky et al. (1997) report overall inter-annotator agreement of 80% kappa, although figures specifically for this tag are not given.

For the BNC-CH and PCC, each transcript is hand-annotated for both self- and other-repair using Healey et al. (2005)'s protocol discussed above. Colman and Healey (2011) and McCabe et al. (2013) report inter-annotator agreement of c.75% kappa. BNC-PGH is annotated only for other-repair initiation P2NTRIs (Purver et al. (2003) report 75%-95% kappa); MAPTASK similarly provides information on P2NTRIs (via `check` tags) but not self-repair.

SWBD, BNC and MAPTASK provide gold-standard part-of-speech (POS) tags; we tagged the PCC using the Stanford POS tagger (Toutanova, Klein, Manning, & Singer, 2003). This is trained on written text; application to spoken dialogue has shown c.10% error rates (Mieskes & Strube, 2006). Here, however, we are not concerned with POS labels *per se*, but in the parallelism between POS sequences - as errors are likely to be fairly consistent (dependent on transcription spelling or spoken dialogue idiosyncracies) we take this as sufficient for our purposes.

Detecting Other-Repair

In order to detect NTRIs, we define a set of turn-level features that could be extracted from transcripts automatically, and that encode either specific NTRI characteristics (e.g. presence of clarificational words like "pardon") or more general parallelism features between the turn to be classified and the previous turn by other and same speaker. (This assumes antecedents of clarification are in the immediately preceding turn; Purver et al. (2003) found this to cover 85% of cases). We then train a standard supervised discriminative classifier using these features to detect NTRI turns.

This approach meets all our requirements. The notion of incrementality here is at the level of speaker turns: we therefore use only information from current and previous turns so that a classification decision can be made immediately (although subsequent turns can certainly contain useful information). Parallelism with context was captured by designing suitable features: lexical parallelism via simple word string matching; syntactic parallelism by matching part-of-speech tags; and semantic parallelism via neural network models of word similarity. Sparsity varies considerably between datasets:

while 11% of MAPTASK utterances are NTRIs, this drops to 4% in the two BNC datasets, 1% in PCC, and only 0.2% in SWBD. To deal with it, we trained the classifier with a weighted cost function, weighting errors on true positive examples more than those on negative ones. Full details of feature calculation and classifier implementation are in the Supplementary Material; the full set of features is shown here in Table 1.

[Table 1 about here]

Results

We test this approach on each of our datasets using 10-fold cross-validation (see Supplementary Material for full details); results are shown in Table 2. Performance is shown against two baselines: always predicting the NTRI class, and using a one-rule classifier with the most helpful single feature. We show performance using our general NTRI and parallelism features ("high-level" features in Table 2), and using all observed unigrams (unique single words, "all" in Table 2). This latter approach illustrates the performance achievable with specific lexical information, but is likely to be highly dataset- and domain-dependent and susceptible to over-fitting, so we treat it as an indicative "ceiling" rather than a suggested robust approach. We also show the performance achieved by Howes, Purver, McCabe, Healey, and Lavelle (2012) on patient-only NTRIs within the PCC dataset, for comparison.

Our primary evaluation metrics are F-score (the harmonic mean of precision and recall) for the class of interest (NTRIs), and the area under the precision-recall curve (AUC-PRC): as our weighted classifiers can be adjusted to trade precision against recall, this AUC metric is more informative than F-score alone; and the F-score we show is for the point where precision and recall are balanced. We also show the more familiar receiver-operator curve area (AUC-ROC), although it is less suitable for unbalanced data, as it underestimates the effect of poor performance on the sparser (and here, more interesting) class (see Saito & Rehmsmeier, 2015).

[Table 2 about here]

Performance varies with the nature of the dataset: with the open-domain BNC, performances are fairly good with F-scores of 52-55% (AUC-PRC 0.51-0.52); in the more domain-specific clinical PCC, F-scores drop below 50%; and in MAPTASK even further to 38%. (Note that baseline F-scores with such unbalanced data are low, with AUC-PRC scores all below 0.22). Encouragingly, the approach seems fairly robust to sparsity itself, with reasonable performance in both the PCC and more open-domain (but telephone-based) SWBD, where NTRIs make up only 1.3% and 0.2% of utterances respectively. (The lowest performance is in the least sparse data (MAPTASK), in fact).

In most datasets, the general high-level features transfer well across domains, with performance similar to the specific unigram features; the exception is MAPTASK and (to a lesser degree) SWBD, suggesting the presence of more domain-specific and/or variable repair mechanisms in those settings. We investigate the most predictive features (by selecting based on information gain); details and feature lists are in the Supplementary Material (Table 6). The most informative are usually the simpler features (interrogative features such as wh-words and question marks; repair keywords; utterance length). Semantic parallelism features (word vector-based similarities) then feature strongly, mixed with the lexical and POS repetition features. However, removing these semantic parallelism features makes little difference to performance: while AUC-PRC tends to drop, indicating less robust performance, the drop is small (1-2%), and the point F-scores do not change; this suggests that the vector-based features capture little information beyond the simpler symbolic ones. Best features for the worst-performing dataset (MAPTASK) are noticeably different, again suggesting different repair mechanisms, with backchannel keywords and repetition seeming to play a stronger role, and wh-words not being useful.

Error analysis. To investigate the limitations and common sources of error, we trained and tested a version on the same full dataset (BNC-PGH), thus giving an upper bound to performance using this feature set. Performance improved only slightly (F=0.54, vs 0.52 using cross-validation), showing that significant limitations exist, and qualitative manual inspection of the errors revealed some common sources of these. NTRI cue words, wh-words, short questions (cued by transcribed question marks) and repetition are all strong indicators, leading to many true positives (19), but are the main cause of false positives (20), (21) (shown ***bold italic***):

(19)[18]

Unknown:	As most of the main towns in Suffolk have reviews every two years are you contemplating having er those, that sort of interview of erm public hearing.
Guy:	**Er what every two years sorry?**
Unknown:	They have traffic management erm reviews every two years.

(20)[19]

e bust:	If it's no I think what we agreed Glynis if it was going to be a stone it could go in the wall where it could be seen from outside.
g herbert:	***Oh right yes sorry I beg your pardon*** .
e bust:	But if we were deciding on a brass plaque or something

(21)[20]

Neal:	`<pause>` Same thing as as I mentioned before. It all fell out. Bags.
Unknown:	`<laugh>`.
Unknown:	***`<unclear>`***?
Neal:	Yes, certainly. She'd got all the clothes she'd ever had.

Omission of question marks in transcription can therefore also cause false negatives. Other false negatives give more interesting insight about what our features fail to capture. In some cases, the key parallelism is not captured by simple sequence and vector-similarity approaches (22); even more challenging are examples with no explicit parallel elements, e.g. P2NTRIs which offer elaborating material (23) or possible continuations (24) (in what Purver et al. (2003) call *gap filler CRs*).

(22)[21]

Anon 1:	Four.
Malcolm:	Yep.
Anon 1:	Six. Nine.
Malcolm:	**<tut> How many ?**
Anon 1:	<unclear> <pause> Nine.
Malcolm:	Nine.

[18]BNC file KN3, sentences 299–301.
[19]BNC file KM8, sentences 599–601.
[20]BNC file KNC, sentences 1075–1080.
[21]BNC file KND, sentences 567–573.

(23)[22]

e bust:	Have have you found out any more <u>the cost</u> Harry of this?
h rickett:	**Yeah for a stone that is ?**
e bust:	Yes.

(24)[23]

e bust:	Ruby `<unclear>` she'll have she'll have some children though because I mean they're somewhere down in <u>. . .</u>
d kemp:	`<unclear>`
e bust:	they're somewhere down in Gillingham down in <u>. . .</u>
d kemp:	**Kent**
e bust:	Yeah they're down in Kent.

Detecting Self-Repair

For self-repair detection we use STIR ('STrongly Incremental Repair detection') (Hough & Purver, 2014).[24] STIR takes a local, incremental approach, detecting the structure in (10) and isolated edit terms (such as 'uh', 'um' and 'you know'), assigning appropriate structural labels – see Figure 1. While sparsity is handled similarly to our other-repair experiments, we now generalise the approach to parallelism: instead of using specific syntactic or semantic knowledge from POS taggers or word vectors, STIR uses a range of information-theoretic measures to capture parallelism in a more general fashion. The notion of incrementality is also different, as a fully word-by-word approach is required (as discussed above).

[Figure 1 here]

Rather than detecting the repair structure in its left-to-right string order, detection consists of 4 time-steps as words are encountered: STIR first detects edit terms (possibly interregna) at step T1; then repair onsets rp_{start} at T2; if one is found, it searches backwards to find the reparandum start rm_{start} at T3; then finally finds the repair end rp_{end} at T4. Step T1 relies mainly on lexical probabilities; T2 exploits features of divergence from 'fluent' language; T3 uses fluency of utterances without the hypothesised reparanda, and parallelism between repair and reparandum; and T4 the similarity between distributions after reparandum and repair end points (indicated by

[22]BNC file KM8, sentences 534–536.

[23]BNC file KM8, sentences 741–744; ellipsis . . . added to show putative 'antecedent'.

[24]Available from http://bitbucket.org/julianhough/stir.

the dotted edge between S3 and S4 in Figure 1). Each stage implements these insights via multiple related features in a statistical classifier, and the four stages are connected together in a pipeline (Figure 2). The output is a graph-like structure (Figure 1). STIR has previously been applied to SWBD; here, we investigate its transfer to our other datasets, and the nature of its errors, while updating it to handle cut-off words.

[Figure 2 about here]

Classifiers and features

Each individual classifier has its own error function, allowing trade-off of immediate accuracy, run-time and stability, and balance in the face of sparsity. Each classifier also uses its own specific combination of features, but all derived from basic information-theoretic measures from n-gram language models (LMs). N-gram LMs are easily applied incrementally, require no commitment to any particular grammar formalism, and can be extended to model levels other than the purely lexical, e.g. grammaticality judgements (A. Clark, Giorgolo, & Lappin, 2013). We train our LMs on the standard Switchboard training data, following Johnson and Charniak (2004) by cleaning the data of all edit terms and reparanda, to approximate a 'fluent' LM. We train two such models, one for words and one for POS tags;[25] this allows us to derive features giving syntactic as well as lexical information, both by using POS tags directly and via A. Clark et al. (2013)'s Weighted Mean Log (*WML*) measures which factor out lexical probability to approximate syntactic plausibility. From these basic LMs we then derive features that characterise (dis)fluency, via probability and *surprisal* for observed words; uncertainty in a context, via the *entropy* of possible continuations, and increases and reductions therein; and similarity or parallelism between contexts, via the Kullback-Leibler (KL) divergence between distributions. We handle partial words within the LM scoring itself, assigning penalties when partial words are encountered. Full details of feature calculation and classifier implementation are given in the Supplementary Material; we give a brief overview here.

[25]Below, measures from the word LM are indicated by the superscript ^{lex} and the POS LM by ^{pos}. When referring to the same measure from both LMs, these are suppressed.

Edit term detection. The first classifier uses the word surprisal s^{lex} from a specific edit word bigram LM (edit words will have high probability and therefore lower s^{lex}), and the trigram surprisal s and syntactic fluency *WML* from the standard fluent LMs described above (the intuition here being that general fluency will seem lower for trigrams containing an edit term). This also helps interregnum recognition, due to the inclusion of interregnum vocabulary within edit term vocabulary (Hough & Purver, 2013), and provides a useful feature for repair detection in subsequent steps (Hough & Purver, 2014; Lease, Johnson, & Charniak, 2006).

Repair start detection. The second step to detect rp_{start} is arguably the most crucial component: the greater its accuracy, the better the input for downstream components and the lesser the overhead of filtering false positives required. This classifier uses a combination of simple alignment features (e.g. whether a word is identical to a predecessor), and a series of features describing local changes in LM fluency. Figure 3 shows the main intuition: that repair onsets correspond to troughs in lexical and syntactic probability measures (in Figure 3, WML^{lex}).

[Figure 3 about here]

Reparandum start detection. We now detect rm_{start} positions given a hypothesised rp_{start}, using two main intuitions. First, we use the noisy channel intuition of Johnson and Charniak (2004) that removing the reparandum (from rm_{start} to rp_{start}) increases fluency of the utterance (captured via *WML* features), while removing non-reparandum words decreases it. Second, we can measure parallelism between rp_{start} and rm_{start}, via the KL divergence between their LM distributions.

Repair end detection and structure classification. Finally, detection of rp_{end} and the final structure of the repair exploits the notion of parallelism. This can be measured as divergence between the conditional probability distributions θ^{lex} at the reparandum-final word rm_{end} and the repair-final word rp_{end}: for repetition repairs, KL divergence will trivially be 0; for substitutions, it will be higher; for deletes, even higher. It can also be captured via *ReparandumRepairDifference*, the difference in probability between an utterance cleaned of the reparandum and the utterance with its repair phase substituting its reparandum. In the running example from Figure 1, this would be as in equation (1).

$$\begin{aligned} &\mathit{ReparandumRepairDifference}(\text{“John [likes + loves]"}) = \\ &\qquad WML^{lex}(\text{“John loves"}) - WML^{lex}(\text{“John likes"}) \end{aligned} \tag{1}$$

Results

Hough and Purver (2014) show state-of-the-art performance for incremental self-repair detection (77.9% accuracy at detecting reparandum words in Switchboard test data); they removed cut-off words which on average occur every 118 words (0.84% of all words) in the Switchboard heldout data. Here we test with cut-off words included, a realistic approach for transcripts and incremental ASR output, and potentially providing further cues about repair onset. By way of comparison, we also test the performance of Hough and Schlangen (2015)'s Recurrent Neural Network (RNN)-based disfluency detector.[26] In all cases, we derive LM features from the SWBD training set using 10-fold cross-validation (full details in the Supplementary Material); we then train and test classifiers using a standard training/test split for each corpus.

We report accuracy of repair onset detection on a per-utterance level, as that is the most relevant measure for dialogue-level analysis; we also report the overall Spearman's rank correlation of the repair rate (per utterance) between the gold standard transcripts and STIR's output. These allow comparison with the PCC and BNC-CH annotations, which use a different annotation schema from Switchboard (see above), and (for BNC-CH) do not mark repair onset point. For Switchboard, we also report the standard per-word reparandum detection result (F *rm*), in line with previous work– see Table 3. This per-word evaluation tells us about ability to identify the precise location of repairs, important for dialogue system development; but the per-utterance figures also give us a useful, if less precise, metric for practical applications such as the analysis of patient-doctor dialogues.

[26]Code available from https://github.com/dsg-bielefeld/deep_disfluency

On Switchboard, accuracy of reparandum word detection reaches 78.1% on the test set, and per-utterance detection accuracy is 85.0%. The correlation for repair rates is very high and significant (*Spearman's rank*=0.956). This marginally improves over Hough and Purver (2014)'s results with partial words removed; and training and testing on the SWBD data with partial words removed in our experimental setup reduces accuracy even more, to 76.8%. This shows the potential utility (rather than hindrance) of using partial words for disfluency detection if adapted appropriately. The RNN model, which is not adapted for partial words, shows the opposite pattern, dropping from 66.8% to 63.8% when *introducing* partial words – see Table 4.

We also test on the out-of-domain PCC and BNC-CH datasets. With PCC, per-utterance detection performance is very encouraging even with no optimization (62.0%), and correlation of repair rates to the gold standard is also high (*Sp. R*=0.805). For BNC-CH, per-utterance results are far worse (41.7%) — we attribute this to the annotation protocol, which lacked the exact identification of reparandum and repair phases used in the other two corpora — however, the correlation of repair rates is still moderately strong (*Sp. R*=0.583, $p<0.001$). Table 3 shows that using POS LM features helps detection performance in each corpus, particularly boosting correlation score for our most challenging dataset, PCC (0.583 vs. 0.530); this suggests that syntactic-level information can help detect repair structures.

Error analysis. The detailed Switchboard annotation format permits a quantitative analysis of the error distribution, and comparison between STIR and the comparable RNN model. Table 5 (a) shows the F-score with different combined reparandum and interregnum lengths, where correct detection is counted if both repair onset and reparandum onset are predicted correctly. All three systems show reduced performance as length increases. However, reduction is less for STIR; its explicit backwards search mechanism alleviates the problem of long-distance dependency, while the RNN relies on internally learned memory structure and struggles further than 5 words back from the repair onset. Table 5 (b) shows performance for different repair types. Repetitions are the easiest, followed by substitutions, then deletes; but STIR performs far better on substitutions and deletions than the RNN. Both of these rarer types rely on more complex notions of parallelism and fluency, rather than the presence of verbatim repeats.

[Tables in Table 5 about here]

A qualitative survey of the errors when changing domain shows that many are due to the transcription and annotation protocols (as discussed by Howes, Hough, Purver, & McCabe, 2014), not merely poor system performance. As shown in examples (25)-(27)[27] from the PCC, false positives occur when STIR tags embedded repairs as multiple instances, but the annotator views this as part of one longer repair (25). False negatives include confusion between editing phrases and repairs (26), a distinction in SWBD but not in Healey et al. (2005)'s annotation protocol; and missing repairs entirely (27), as utterance-initial deletions are not marked in SWBD but treated as separate utterances.

(25)[28]
(a) D: . . . and if you tell me that **that**[RP_{START}] that the depressions kicks in . . .
(b) D: . . . and if you tell me that **that**[rp_{start}] **that**[rp_{start}] the depressions kicks in . . .

(26)[29]
(a) D: and so **I**[RP_{START}] mean otherwise I'm not too concerned about your mental health. . .
(b) D: and so **I**[ed] **mean**[ed] otherwise I'm not too concerned about your mental health. . .

(27)[30]
(a) P: I don't **I'm**[RP_{START}] not like hearing voices. . .
(b) P: I don't I'm not like hearing voices. . .

Discussion and Conclusions

Our experiments show that detection of both self-repair and other-repair initiation is possible with reasonable accuracy. For the self-repair case, by-utterance F-scores can reach 85% when trained on in-domain data, and up to 62% even when transferring a model to other (here, face-to-face clinical) data. For the much sparser other-repair case, F-scores can reach 60%, but depend on the nature of the data; while robust to sparsity

[27] Hand annotation tags are shown in (a) in each case with STIR's annotations shown in (b).
[28] (Howes et al., 2014) example (10)
[29] (Howes et al., 2014) example (11)
[30] (Howes et al., 2014) example (12)

itself in Switchboard where NTRIs are particularly sparse (0.2% of turns), some domains cause bigger drops, although in the sparse clinical data F-scores still reach 46%. These results are encouraging as they use general models which exploit features of repair-indicating vocabulary and parallelism, hence giving robustness across datasets and being applicable to the general case of third-party dialogue analysis.

Examination of the effect of features suggests that the key to good performance is capturing parallelism, reflecting the nature of repair as a resource for querying and reformulating material. However, this seems hard to achieve using general models of word meaning (as in our other-repair classifier): using general lexical matching and suitably trained information-theoretic models of word distributions, as STIR does for self-repair, seems more successful, and more robust across domains and phenomena than more directly lexically driven approaches (here, the comparison RNN). A possible direction for future research would be to investigate whether similar methods could help with the challenging cases of implicit parallelism seen with other-repair.

The effect of changing domains and genres suggests that some domains show different repair phenomena and mechanisms. Inspection of the task-driven Map Task data shows that the challenging other-repair types are more common (e.g. offering elaboration and reformulation), as is long-range clarification, where participants check their understanding of whole sequences of instructions (rare in the other datasets). Many of the domain-related effects, though, are associated with differences in transcription and annotation standards, as discussed above for self-repair. This is also a factor with other-repair data; for example, the Map Task annotations tag some forms of NTRI question as belonging instead to an 'other question' category (28).

(28)[31]
G: until you you get over the top of the <u>slate mountain</u>
F: **over the top of the**
G: slate mountain
F: don't have a slate mountain

However, in many cases these differences in annotation approach stem from genuine ambiguity or multifunctionality. We have seen cases of self-repair where

[31]Map Task corpus, dialogue q1ec2, utterances 59-62.

alternate analyses are possible (25)-(27), cases of other-repair which perform repair initiation simultaneously with offering possible repair (23)-(24), and many forms (e.g. repeated fragments) can also perform acknowledgement or answer questions. Recognising and handling this ambiguity is of course crucial for dialogue systems, although resolving it is not always possible or desirable — hence the success of probabilistic models which maintain uncertainty (Young et al., 2013) — and this suggests that repair identification should be approached and evaluated in a probabilistic fashion, not a categorical one. This also points to the limitations of using transcripts as our source material. For human annotators, one of the signals of an NTRI is whether the *following* turn contains a *position 3 other initiated self repair* – i.e. whether the other dialogue participant has interpreted the preceding turn as requesting repair; our incremental approach means we cannot benefit from this information. Of course, participants in dialogue must decide whether to treat a turn as initiating repair as and when they encounter it – so this cannot be how humans identify these whilst engaged in dialogue. Evidence suggests that in real dialogue, feedback (positive or negative) is cued by or accompanied by gaze (Hjalmarsson & Oertel, 2012), intonation (Gravano & Hirschberg, 2009) or gesture (Healey et al., 2013; Healey, Plant, Howes, & Lavelle, 2015), suggesting that we may improve our performance if we include these features.

Despite these limitations, these models go a long way toward fulfilling our desiderata: they operate *incrementally* (utterance-by-utterance for P2NTRIs, word-by-word for P1SISRs) and *monotonically* (STIR leaves reparandum material available for later processing); they use general measures of *parallelism with context*; and they are relatively robust to the *sparsity* of NTRIs and rarer and longer self-repairs. Such models therefore have potential not only to help make human-computer dialogue systems more human-like, via more robust, incremental self-repair and other-repair detection; but also to improve our ability to analyse and evaluate the quality of communication in settings like clinical psychiatry.

Acknowledgements

Purver was partially supported by EPSRC (grant EP/J010383/1) and by the ConCreTe project, which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733. Hough was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, funded by the German Research Foundation (DFG), and the DFG-funded DUEL project (grant SCHL 845/5-1). Howes was supported by two grants from the Swedish Research Council (VR); 2016-0116 – Incremental Reasoning in Dialogue (IncReD) and 2014-39 for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg. We thank David Schlangen for extensive discussions on the topic.

References

Anderson, A., Bader, M., Bard, E., Boyle, E., Doherty, G., Garrod, S., ... Weinert, R. (1991). The HCRC map task data. *Language and Speech*, *34*(4), 351–366.

Bard, E. G., Lickley, R. J., & Aylett, M. P. (2001). Is disfluency just difficulty? In *Isca tutorial and research workshop (itrw) on disfluency in spontaneous speech.*

Bortfeld, H., Leon, S. D., Bloom, J. E., Schober, M. F., & Brennan, S. E. (2001). Disfluency rates in conversation: Effects of age, relationship, topic, role, and gender. *Language and speech*, *44*(2), 123–147.

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

Burnard, L. (2000). *Reference guide for the british national corpus (world edition).* Oxford University Computing Services. Retrieved from http://www.natcorp.ox.ac.uk/docs/userManual/

Clark, A., Giorgolo, G., & Lappin, S. (2013, August). 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)* (pp. 28–36). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/W13-2604

Clark, H. H. (1996). *Using language.* Cambridge University Press.

Colman, M., & Healey, P. G. T. (2011). The distribution of repair in dialogue. In L. Carlson, C. Hoelscher, & T. F. Shipley (Eds.), *Proceedings of the 33rd annual meeting of the cognitive science society* (pp. 1563–1568). Boston, MA: Cognitive Science Society.

Dingemanse, M., Roberts, S. G., Baranova, J., Blythe, J., Drew, P., Floyd, S., ... others (2015). Universal principles in the repair of communication problems. *PloS one*, *10*(9), e0136100. Retrieved from http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0136100

Dingemanse, M., Torreira, F., & Enfield, N. J. (2013). Is "Huh?" a universal word? Conversational infrastructure and the convergent evolution of linguistic items.

PloS one, *8*(11), e78273. Retrieved from http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0078273

Fernández, R., Ginzburg, J., & Lappin, S. (2007). Classifying ellipsis in dialogue: A machine learning approach. *Computational Linguistics*, *33*(3), 397–427.

Ferreira, F., Lau, E. F., & Bailey, K. G. D. (2004). Disfluencies, language comprehension, and tree adjoining grammars. *Cognitive Science*, *28*(5), 721ñ749.

Ginzburg, J., & Cooper, R. (2004). Clarification, ellipsis, and the nature of contextual updates in dialogue. *Linguistics and Philosophy*, *27*(3), 297–365.

Ginzburg, J., Fernández, R., & Schlangen, D. (2007). Unifying self- and other-repair. In R. Artstein & L. Vieu (Eds.), *Proceedings of the 11th workshop on the semantics and pragmatics of dialogue (DECALOG).* Rovereto, Italy: SemDial.

Godfrey, J. J., Holliman, E., & McDaniel, J. (1992). SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of IEEE ICASSP-92* (pp. 517–520). San Francisco, CA: IEEE.

Goodwin, C. (1979). The interactive construction of a sentence in natural conversation. In G. Psathas (Ed.), *Everyday language: Studies in ethnomethodology* (pp. 97–121). New York: Irvington Publishers.

Gravano, A., & Hirschberg, J. (2009). Backchannel-inviting cues in task-oriented dialogue. In M. Uther, R. Moore, & S. Cox (Eds.), *Interspeech* (pp. 1019–1022). Brighton, UK: ISCA.

Healey, P. G. T., Colman, M., & Thirlwell, M. (2005). Analysing multi-modal communication: Repair-based measures of human communicative co-ordination. In J. van Kuppevelt, L. Dybkjaer, & N. Bernsen (Eds.), *Natural, intelligent and effective interaction in multimodal dialogue systems* (Vol. 30, pp. 113–129). Dordrecht: Kluwer.

Healey, P. G. T., Lavelle, M., Howes, C., Battersby, S., & McCabe, R. (2013, July). How listeners respond to speaker's troubles. In M. Knauff, M. Pauen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th annual conference of the cognitive science society* (pp. 2506–2511). Berlin: Cognitive Science Society.

Healey, P. G. T., Plant, N., Howes, C., & Lavelle, M. (2015). When words fail: Collaborative gestures during clarification dialogues. In S. Andrist, D. Bohus, E. Horvitz, B. Mutlu, & D. Schlangen (Eds.), *2015 aaai spring symposium series: Turn-taking and coordination in human-machine interaction* (pp. 23–29). Stanford, CA: AAAI Press.

Hjalmarsson, A., & Oertel, C. (2012). Gaze direction as a back-channel inviting cue in dialogue. In J. Edlund, I. de Kok, R. Poppe, & D. Traum (Eds.), *Iva 2012 workshop on realtime conversational virtual agents* (Vol. 9). Santa Cruz, CA.

Honnibal, M., & Johnson, M. (2014). Joint incremental disfluency detection and dependency parsing. *Transactions of the Association of Computational Linguistics (TACL)*, *2*, 131-142.

Hough, J. (2015). *Modelling incremental self-repair processing in dialogue* (Unpublished doctoral dissertation). Queen Mary University of London.

Hough, J., & Purver, M. (2012, September). Processing self-repairs in an incremental type-theoretic dialogue system. In S. Brown-Schmidt, J. Ginzburg, & S. Larsson (Eds.), *Proceedings of the 16th SemDial workshop on the semantics and pragmatics of dialogue (SeineDial)* (pp. 136–144). Paris, France: SemDial.

Hough, J., & Purver, M. (2013, December). Modelling expectation in the self-repair processing of annotat-, um, listeners. In R. Fernández & A. Isard (Eds.), *Proceedings of the 17th SemDial workshop on the semantics and pragmatics of dialogue (DialDam)* (pp. 92–101). Amsterdam: SemDial.

Hough, J., & Purver, M. (2014). Strongly incremental repair detection. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp).* Doha, Qatar: Association for Computational Linguistics.

Hough, J., & Schlangen, D. (2015). Recurrent neural networks for incremental disfluency detection. *Interspeech 2015*.

Howes, C., Hough, J., Purver, M., & McCabe, R. (2014, September). Helping, i mean assessing psychiatric communication: An application of incremental self-repair detection. In V. Rieser & P. Muller (Eds.), *Proceedings of the 18th SemDial workshop on the semantics and pragmatics of dialogue (DialWatt)* (pp. 80–89). Edinburgh: SemDial.

Howes, C., Lavelle, M., Healey, P. G. T., Hough, J., & McCabe, R. (2017). Disfluencies in dialogues with patients with schizophrenia. In G. Gunzelmann, A. Howes, T. Tenbrink, & E. Davelaar (Eds.), *Proceedings of the 39th annual meeting of the cognitive science society.* London, UK: Cognitive Science Society.

Howes, C., Purver, M., Healey, P. G. T., Mills, G. J., & Gregoromichelaki, E. (2011). On incrementality in dialogue: Evidence from compound contributions. *Dialogue & Discourse*, *2*(1), 279–311. Retrieved from http://dad.uni-bielefeld.de/index.php/dad/article/view/362 doi: 10.5087/dad.2011.111

Howes, C., Purver, M., McCabe, R., Healey, P. G. T., & Lavelle, M. (2012). Helping the medicine go down: Repair and adherence in patient-clinician dialogues. In S. Brown-Schmidt, J. Ginzburg, & S. Larsson (Eds.), *Proceedings of the 16th SemDial workshop on the semantics and pragmatics of dialogue (SeineDial)* (pp. 155–156). Paris: SemDial.

Johnson, M., & Charniak, E. (2004). A tag-based noisy channel model of speech repairs. In *Proceedings of the 42nd annual meeting on association for computational linguistics.* Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dx.doi.org/10.3115/1218955.1218960 doi: http://dx.doi.org/10.3115/1218955.1218960

Jurafsky, D., Shriberg, E., & Biasca, D. (1997). *Switchboard swbd-damsl shallow-discourse-function annotation coders manual, draft 13* (Tech. Rep. No. 97-02). University of Colorado, Boulder. Institute of Cognitive Science.

Kitaoka, N., Kakutani, N., & Nakagawa, S. (2005). Detection and recognition of correction utterances on misrecognition of spoken dialog system. *Systems and Computers in Japan*, *36*(11), 24–33. Retrieved from http://dx.doi.org/10.1002/scj.20341 doi: 10.1002/scj.20341

Lake, J. K., Humphreys, K. R., & Cardy, S. (2011). Listener vs. speaker-oriented aspects of speech: Studying the disfluencies of individuals with autism spectrum disorders. *Psychonomic bulletin & review*, *18*(1), 135–140.

Lease, M., Johnson, M., & Charniak, E. (2006). Recognizing disfluencies in conversational speech. *Audio, Speech, and Language Processing, IEEE Transactions on*, *14*(5), 1566–1573.

Lemon, O., & Gruenstein, A. (2004). Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction*, *11*(3), 1–27.

Leudar, I., Thomas, P., & Johnston, M. (1992). Self-repair in dialogues of schizophrenics: Effects of hallucinations and negative symptoms. *Brain and Language*, *43*(3), 487 - 511.

Levelt, W. (1983). Monitoring and self-repair in speech. *Cognition*, *14*(1), 41–104.

Levelt, W. (1989). *Speaking: From intention to articulation.* MIT Press.

Lickley, R. J. (2001). Dialogue moves and disfluency rates. In *Isca tutorial and research workshop (itrw) on disfluency in spontaneous speech* (pp. 93–96). Edinburgh: ISCA.

Litman, D., Hirschberg, J., & Swerts, M. (2006). Characterizing and predicting corrections in spoken dialogue systems. *Computational Linguistics*, *32*(3), 417–438.

Lopes, J., Salvi, G., Skantze, G., Abad, A., Gustafson, J., Batista, F., … Trancoso, I. (2015, September). Detecting repetitions in spoken dialogue systems using phonetic distances. In S. Möller, H. Ney, B. Möbius, E. Nöth, & S. Steid (Eds.), *Proceedings of interspeech* (pp. 1805–1809). Dresden: ISCA.

McCabe, R. (2008). *Doctor-patient communication in the treatment of schizophrenia: Is it related to treatment outcome?* (Tech. Rep.). Final report on G0401323 to Medical Research Council.

McCabe, R., Healey, P. G. T., Priebe, S., Lavelle, M., Dodwell, D., Laugharne, R., … Bremner, S. (2013). Shared understanding in psychiatrist-patient communication:

Association with treatment adherence in schizophrenia. *Patient Education and Counselling*.

Meteer, M., Taylor, A., MacIntyre, R., & Iyer, R. (1995). *Disfluency annotation stylebook for the Switchboard Corpus* (Tech. Rep.). Department of Computer and Information Science, University of Pennsylvania. Retrieved from ftp://ftp.cis.upenn.edu/pub/treebank/swbd/doc/DFL-book.ps

Mieskes, M., & Strube, M. (2006). Part-of-speech tagging of transcribed speech. In N. Calzolari et al. (Eds.), *Proceedings of lrec* (pp. 935–938). Genoa: LREC.

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of naacl-hlt* (pp. 746–751).

Mills, G. (2013, December). Establishing a communication system: Miscommunication drives abstraction. In R. Fernández & A. Isard (Eds.), *Proceedings of the 17th SemDial workshop on the semantics and pragmatics of dialogue (DialDam).* Amsterdam: SemDial.

Mills, G., & Healey, P. G. T. (2006, September). Clarifying spatial descriptions: Local and global effects on semantic co-ordination. In *Proceedings of the 10th workshop on the semantics and pragmatics of dialogue (SEMDIAL).* Potsdam, Germany.

Ong, L., De Haes, J., Hoos, A., & Lammes, F. (1995). Doctor-patient communication: a review of the literature. *Social science & medicine*, *40*(7), 903–918.

Oviatt, S. (1995). Predicting spoken disfluencies during human–computer interaction. *Computer Speech & Language*, *9*(1), 19–35.

Purver, M., Ginzburg, J., & Healey, P. G. T. (2003). On the means for clarification in dialogue. In R. Smith & J. van Kuppevelt (Eds.), *Current and new directions in discourse & dialogue* (pp. 235–255). Kluwer Academic Publishers.

Rasooli, M. S., & Tetreault, J. (2014). Non-monotonic parsing of fluent umm I mean disfluent sentences. In S. Wintner, S. Goldwater, & S. Riezler (Eds.), *Eacl 2014* (pp. 48–53). Gothenburg: Association for Computational Linguistics.

Raux, A., Langner, B., Black, A., & Eskenazi, M. (2005). Let's go public! taking a spoken dialog system to the real world. In I. Trancoso (Ed.), *Proceedings of* *interspeech 2005 (eurospeech).* Lisbon, Portugal: ISCA.

Rieser, V., & Moore, J. (2005, June). Implications for generating clarification requests in task-oriented dialogues. In K. Knight, H. T. Ng, & K. Oflazer (Eds.), *Proceedings of the 43rd annual meeting of the ACL* (pp. 239–246). Ann Arbor.

Rodríguez, K., & Schlangen, D. (2004, July). Form, intonation and function of clarification requests in German task-oriented spoken dialogues. In J. Ginzburg & E. Vallduví (Eds.), *Proceedings of the 8th workshop on the semantics and pragmatics of dialogue (SEMDIAL)* (pp. 101–108). Barcelona, Spain: SemDial.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, *10*(3), e0118432. Retrieved from https://doi.org/10.1371/journal.pone.0118432

San-Segundo, R., Montero, J. M., Ferreiros, J., Córdoba, R., & Pardo, J. M. (2001, September). Designing confirmation mechanisms and error recover techniques in a railway information system for Spanish. In D. Traum, J. Hirschberg, R. Smith, & J. van Kuppevelt (Eds.), *Proceedings of the 2nd SIGdial workshop on discourse and dialogue* (pp. 136–139). Aalborg, Denmark: Association for Computational Linguistics.

Schegloff, E., Jefferson, G., & Sacks, H. (1977). The preference for self-correction in the organization of repair in conversation. *Language*, *53*(2), 361–382.

Schlangen, D. (2005, June). Towards finding and fixing fragments: Using machine learning to identify non-sentential utterances and their antecedents in multi-party dialogue. In K. Knight, H. T. Ng, & K. Oflazer (Eds.), *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL)* (pp. 247–254). Ann Arbor, MI.

Shriberg, E. (1994). *Preliminaries to a theory of speech disfluencies* (Unpublished doctoral dissertation). University of California, Berkeley.

Skantze, G., & Hjalmarsson, A. (2010, September). Towards incremental speech generation in dialogue systems. In *Proceedings of the sigdial 2010 conference* (pp.

1–8). Tokyo, Japan: Association for Computational Linguistics. Retrieved from http://www.sigdial.org/workshops/workshop11/proc/pdf/SIGDIAL01.pdf

Stivers, T., & Enfield, N. J. (2010). A coding scheme for question–response sequences in conversation. *Journal of Pragmatics*, *42*(10), 2620–2626.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., ... Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, *26*(3), 339–373.

Stoyanchev, S., Liu, A., & Hirschberg, J. (2014, April). Towards natural clarification questions in dialogue systems. In *Aisb symposium on questions, discourse and dialogue: 20 years after Making it Explicit.*

Stoyanchev, S., & Stent, A. (2012). Concept type prediction and responsive adaptation in a dialogue system. *Dialogue & Discourse*, *3*(1), 1–31.

Surendran, D., & Levow, G.-A. (2006). Dialog act tagging with support vector machines and hidden Markov models. In R. M. Stern (Ed.), *Proceedings of interspeech.* Pittsburgh, PA: ISCA.

Toutanova, K., Klein, D., Manning, C., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In M. Hearst & M. Ostendorf (Eds.), *Proceedings of HLT-NAACL* (pp. 252–259). Edmonton: Association for Computational Linguistics.

Turian, J., Ratinov, L.-A., & Bengio, Y. (2010, July). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384–394). Uppsala, Sweden: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P10-1040

Weng, F., Yan, B., Feng, Z., Ratiu, F., Raya, M., Lathrop, B., ... Peters, S. (2007, September). CHAT to your destination. In *Proceedings of the 8th SIGdial workshop on discourse and dialogue* (p. 79-86). Antwerp, Belgium. Retrieved from http://godel.stanford.edu/twiki/pub/Public/SemlabPublications/weng-et-al07sigdial.pdf

Young, S., Gašić, M., Thomson, B., & Williams, J. D. (2013, May). POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, *101*(5), 1160-1179. doi: 10.1109/JPROC.2012.2225812

Zwarts, S., Johnson, M., & Dale, R. (2010). Detecting speech repairs incrementally using a noisy channel approach. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 1371–1378). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://portal.acm.org/citation.cfm?id=1873781.1873935

Appendix

Materials for Replication

The PCC corpus is confidential due to its sensitive nature; all other data and experiment processing scripts are publicly available. The scripts for the other-repair experiments can be accessed via the Open Science Framework at http://osf.io/w4dmz; scripts and pre-processed data for the self-repair experiments can be accessed via the git repository http://bitbucket.org/julianhough/stir. The original datasets can be obtained as follows:

- **SWBD**: The original corpus is available from http://www.stanford.edu/~jurafsky/swb1_dialogact_annot.tar.gz; we also used the associated Python package available at http://compprag.christopherpotts.net/swda.html.

- **BNC**: The original corpus is available from http://purl.ox.ac.uk/ota/2554. The BNC-PGH and BNC-CH annotations are included with our experiment scripts on the OSF.

- **MAPTASK**: The original corpus is available from http://groups.inf.ed.ac.uk/maptask/; we used the V2.1 NXT format annotations.

Supplementary Material

Experimental Details: Other-Repair

Our turn-level NTRI classifier uses two categories of features. One is designed to capture characteristic surface properties of NTRIs: presence of wh-words, specific clarification keywords (e.g. "pardon"), and behaviour associated with repair such as fillers, pauses and overlaps. The second is designed to capture parallelism. Lexical parallelism is captured via simple word string matching; syntactic parallelism by matching POS tags. For semantic parallelism, we measure word and turn similarity using distributed vector representations from two neural models, (Mikolov, Yih, & Zweig, 2013)'s word2vec with 300 dimensions trained on the Google News corpus, and Turian, Ratinov, and Bengio (2010)'s implementation of (Collobert & Weston, 2008) with 100 dimensions trained on the Reuters RCV1 corpus. The full set of features is given in Table 1.

For the majority of features, we extracted one raw feature (the numeric count or binary indicator, see Table 1) and one proportional feature (the proportion of the turn made up of the feature in question, from 0 to 1). For vector-based similarity features, we extracted four features (minimum, mean and maximum pairwise word cosine similarities, and overall turn cosine similarity summing word vectors within turns following Mitchell and Lapata (2010)). Parallelism features were calculated between the turn being classified, and the preceding turns by the same and other speaker separately. The features ranked in terms of information gain (using Weka's default implementation with best-first search) are shown in Table 6 – see above for discussion.

We experimented with logistic regression, decision tree and support vector machine (SVM) classifiers, as implemented in the Weka toolkit (Hall et al., 2009). Results given here use SVMs with radial basis function kernels; logistic regression and linear-kernel SVMs performed very similarly for all datasets other than SWBD, while decision trees were usually worse. For each dataset, we used 10-fold cross-validation (using Weka's built-in stratified cross-validation routines): the dataset is randomised and split into 10 equally-sized parts, and the classifier tested on each 10% part in turn while training on the other 90%.

To combat sparsity, we weighted the classifiers to give equal precision and recall for the rare class of interest (here, NTRIs), by either adjusting the decision threshold directly (for logistic regression) or by weighting the rarer class more highly in the training error function (for support vector machines). Unweighted versions gave very low recall due to the rarity of NTRIs. For the latter, we used Weka's built-in class weighting function; the underlying implementations vary for particular classifier types, but share the intuition shown in (2,3). Here, C is a single global cost parameter, I is the set of training instances and ϕ_i the individual error for any given instance under a particular model's prediction; C_+, C_- are weighted cost parameters for positive and negative classes respectively, and I_+, I_- the sets of associated instances.

$$\text{standard error term:} \quad C \sum_{i \in I} \phi_i \tag{2}$$

$$\text{weighted error term:} \quad C_+ \sum_{i \in I_+} \phi_i + C_- \sum_{i \in I_-} \phi_i \tag{3}$$

Both methods allow precision to be traded off against recall, and the best choice in practice will depend on application and aims; we give results for the point where precision and recall are equal, and show the area under the precision-recall curve as a measure which abstracts away from the exact setting.

Experimental Details: Self-Repair

Our word- (and partial-word-)level self-repair classifier uses STIR, with features derived from n-gram language models (LMs). The basic 'fluent' LMs are trigram LMs with Kneser-Ney smoothing (Kneser & Ney, 1995), trained on the standard Switchboard training data cleaned of all edit terms and reparanda, giving a total of ≈100K utterances, ≈600K tokens. We train one LM for words and one for POS tags.[32] We call their probabilities p^{lex} and p^{pos} respectively below; if referring to the same calculation for both models we suppress the superscripts.

[32] In pre-processing, POS tags in a many-to-one relation to words are concatenated into one token; this had no significant effect on results.

From the basic probability values we derive our principal lexical uncertainty measurement *surprisal* s (equation 4); and, following Clark et al. (2013), the (unigram) Weighted Mean Log trigram probability (*WML*, eq. 5) – this factors out lexical frequency to approximate *incremental syntactic probability*.

$$s(w_{i-2} \dots w_i) = -\log_2 p(w_i \mid w_{i-2}, w_{i-1}) \tag{4}$$

$$WML(w_0 \dots w_n) = \frac{\sum_{i=2}^{n} \log_2 p(w_i \mid w_{i-2}, w_{i-1})}{-\sum_{j=2}^{n} \log_2 p(w_j)} \tag{5}$$

$$H(w|c) = -\sum_{w \in Vocab} p(w|c) \log_2 p(w|c) \quad \text{where} \quad c = w_{i-2}, w_{i-1} \tag{6}$$

As a measure of uncertainty, we can then derive the entropy $H(w|c)$ of possible word continuations w given a context c, from $p(w_i|c)$ for all words w_i in the vocabulary – see (6). Calculating distributions over the entire lexicon incrementally is costly, so this is approximated by calculating directly only for words observed at least once in context c in training, assuming a uniform distribution over unseen suffixes (see Hough & Purver, 2014). We can then measure increases and reductions in entropy, and similarity between distributions in two different contexts c_1 and c_2 via the Kullback-Leibler (KL) divergence (relative entropy, using a similar approximation).

Adaptation for processing partial words. We adapt our language model scoring when there is a partial, cut-off word transcribed as the penultimate word in any n-gram (in our case, the second word of any trigram). This captures the idea of the fluency dropping after the cut-off word has been processed. We simply assign such trigrams a minimum probability $\frac{1}{|V|}$ where $\mid V \mid$ is the vocabulary size.

While this simple method gives good results in practice, we have also developed a more principled off-line model. We train a simple word completion model $p^{complete}(w|w_i)$ which operates on any annotated partial word prefix w_i to provide a distribution over possible complete words that it could have started, and thus also the most likely completion (based on the prefix and unigram co-occurrence). This is combined with the language model probability p^{lex} within the function p^{fluent}, which for a partial word w_i, gives the likelihood of a given word w being its corresponding complete word at the time of interruption given its two word context is as in (7).

$$p^{fluent}(w \mid w_{i-2}, w_{i-1}, w_i) = \frac{1}{Z} \times p^{lex}(w \mid w_{i-2}, w_{i-1}) \times p^{complete}(w \mid w_i) \tag{7}$$

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

The probability $p^{\hat{fluent}}$ of the most likely complete word guess for w_i is therefore:

$$p^{\hat{fluent}}(w \mid w_{i-2}, w_{i-1}, w_i) = \max_w p^{fluent}(w \mid w_{i-2}, w_{i-1}, w_i) \tag{8}$$

The intuition here is that when hearers encounter a partial word, they attempt to find the fluent word most likely to both complete the partial word and follow the two preceding words. The probability of a completion 'remember' will be higher after "Yes I remem-" than in a less predictable context e.g. utterance-initial "Re-".

Classifiers and features. The 4 individual classifiers in STIR then use combinations of features derived from these basic measures.

Edit term detection. We use the word surprisal s^{lex} from a specific edit word bigram model (expexting low s^{lex} for words likely to be edit terms), and the trigram surprisal s and syntactic fluency WML from the standard fluent word and POS models described above. The decision task is to classify at the current position w_i, one, both or none of words w_i and w_{i-1} as edit terms. We found this simple approach effective and stable, detecting edit term words with an F-score of 0.938, performing marginally worse though detecting a broader range of phenomena than Heeman and Allen (1999)'s discourse marker detector. Some delayed decisions occur in cases where s^{lex} and WML^{lex} have similar values in both the edit and fluent language models before the end of the edit, e.g. "I like" → "I {*like*} want...", with classification only achieved at w_{i-1}; this could cause some output instability or 'jitter'.

Repair start detection. Starting with s, WML, H for word and POS models, we derive 5 additional information-theoretic features: ΔWML is the difference between the WML values at w_{i-1} and w_i; ΔH is the difference in entropy between w_{i-1} and w_i; $InformationGain$ is the difference between expected entropy at w_{i-1} and observed s at w_n, a measure that factors out the effect of naturally high entropy contexts;

BestEntropyReduce is the best reduction in entropy possible by an early rough hypothesis of reparandum onsets within 3 words; and *BestWMLboost* similarly speculates on the best improvement of *WML* possible by positing rm_{start} positions up to 3 words back. We also include simple alignment features: binary features which indicate if the word w_{i-x} is identical to the current word w_i for $x \in \{1, 2, 3\}$. With 6 alignment features, 16 language model information-theoretic features and a single logical feature *edit* which indicates the presence of an edit word at position w_{i-1}, rp_{start} detection uses 23 features– see Table 7.

Ranking the features by Information Gain using 10-fold cross validation over the Switchboard heldout data (see Table 7) shows that the language model features are far more discriminative than the alignment features, with *WML* in both p^{lex} and p^{pos} models being the most discriminative. Actual lexical or POS *values* (i.e. words and POS tags) are not used at all in the feature sets, only these information-theoretic measures.

Reparandum start detection. Altogether we use 32 features, and again information-theoretic ones are most useful. The two best features capture the noisy channel intuition that removing the reparandum increases fluency: they are Δ*WMLboost* (the drop in *WMLboost* from one backtracked position to the next) for word and POS models. The third best feature measures parallelism between rp_{start} and rm_{start}, via the KL divergence between $\theta^{pos}(w \mid rm_{start}, rm_{start-1})$ and $\theta^{pos}(w \mid rp_{start}, rp_{start-1})$.

Repair end detection and structure classification. Finally, detection of rp_{end} uses parallelism, measured as KL divergence between the conditional probability distribution θ^{lex} at the reparandum-final word rm_{end} and the repair-final word rp_{end}. For repetition repairs, divergence will trivially be 0; for substitutions, it will be higher; for deletes, even higher. It can also be captured via *ReparandumRepairDifference*, the difference in probability between an utterance cleaned of the reparandum and the utterance with its repair phase substituting its reparandum. In the running example from Figure 1, this would be as in equation (9).

$$\mathit{ReparandumRepairDifference}(\text{“John [likes + loves]"}) = \mathit{WML}^{lex}(\text{“John loves"}) - \mathit{WML}^{lex}(\text{“John likes"}) \quad (9)$$

Classifier pipeline and training setup. The classifiers are implemented using Random Forests (Breiman, 2001), using different error functions for each stage via MetaCost (Domingos, 1999); in early investigation this outperformed single decision tree classifiers. The LM-derived features are obtained using a 10-fold cross-validation method, always using the SBWD training set: for each fold, we train the LMs on 90% and use them to calculate feature values on the unseen 10% (this avoids over-fitting probability values). We then use these feature values to train and test the classifiers using a standard single development/test split for each corpus .

In both training and testing, the classifiers are combined in a pipeline as in Figure 2, where the *ed* classifier only permits non-*ed* words to be passed on to rp_{start} classification. The rp_{start} classifier passes positive repair hypotheses to the rm_{start} classifier, which searches backwards up to 7 words which have not been classified as edit terms e. If a rm_{start} is classified, the output is passed on for rp_{end} classification. Active repair hypotheses are added to a stack, each consisting of a $\langle rm_{start}, rp_{start}, rp_{end}\rangle$ triple of word positions; the position of rp_{end} may change as more words are consumed. The rp_{end} detector may temporarily *cancel* a hypothesis after two words have been consumed beyond the repair onset, which does not remove the hypothesis indefinitely but subdues its effect in its output before searching for more suitable rp_{end} points– this could cause output jitter. Repair hypotheses are are popped off the stack when the string is 7 words beyond its rp_{start} position. Putting limits on the stack's storage space is a way of controlling for processing overhead and complexity. Embedded repairs whose rm_{start} coincide with another's rp_{start} are easily dealt with as they are added to the stack as separate hypotheses.[33] In terms of complexity, the number of possible repairs grows approximately in the triangular number series– i.e. $\frac{n(n+1)}{2}$, a nested loop over previous words as n gets incremented, which in terms of a complexity class is a quadratic $O(n^2)$.

[33]We constrain the problem not to include embedded deletes which may share their rp_{start} word with another repair – these are in practice very rare.

References

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Clark, A., Giorgolo, G., & Lappin, S. (2013, August). 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)* (pp. 28–36). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/W13-2604

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167).

Domingos, P. (1999). Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 155–164).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGDKDD Explorations*, *11*(1), 10–18.

Heeman, P., & Allen, J. (1999). Speech repairs, intonational phrases, and discourse markers: modeling speakers' utterances in spoken dialogue. *Computational Linguistics*, *25*(4), 527–571.

Hough, J., & Purver, M. (2014). Strongly incremental repair detection. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp).* Doha, Qatar: Association for Computational Linguistics.

Kneser, R., & Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acoustics, speech, and signal processing, 1995. icassp-95., 1995 international conference on* (Vol. 1, pp. 181–184).

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of naacl-hlt* (pp. 746–751).

Mitchell, J., & Lapata, M. (2010). Composition in distributional models of semantics.

Cognitive Science, *34* (8), 1388–1439.

Turian, J., Ratinov, L.-A., & Bengio, Y. (2010, July). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384–394). Uppsala, Sweden: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P10-1040

Measure	Description
NumWords	Number of words in turn
OpenClassRepair	Number of Open Class Repair Initiator words (e.g. *pardon, huh*)
WhWords	Number of wh-words (e.g. *what, who, when*)
Backchannel	Number of backchannels (e.g. *uh-huh, yeah*)
FillerWords	Number of fillers (e.g. *er, um*)
MarkedPauses	Number of pauses transcribed
OverlapAny	Number of portions of overlapping talk
OverlapAll	Entirely overlapping another turn
RepeatedWords	Number of words repeated from preceding turn
RepeatedPos	Number of PoS-tags repeated from preceding turn
W2vSim	Cosine similarity with preceding turn (word2vec, Mikolov et al., 2013)
TeaSim	Cosine similarity with preceding turn (Turian et al., 2010)

Table 1

Turn-level features for NTRI detection

Corpus	Features	P	R	F	AUC-PRC	AUC-ROC
PCC patient	OCRProportion	.86	.23	.36	-	-
PCC patient	High-level	.43	.41	.41	-	-
PCC patient	All	.45	.44	.44	-	-
PCC all	(baseline)	.01	1.0	.03	.01	.50
PCC all	qMarkProportion	.65	.14	.24	.11	.57
PCC all	High-level	.44	.43	.44	.40	.90
PCC all	All	.46	.47	.46	.43	.73
BNC-CH	(baseline)	.04	1.0	.08	.04	.49
BNC-CH	qMarkProportion	.62	.31	.41	.22	.65
BNC-CH	High-level	.55	.55	.55	.52	.90
BNC-CH	All	.57	.62	.60	.61	.80
BNC-PGH	(baseline)	.04	1.0	.08	.04	.50
BNC-PGH	OCRProportion	.70	.09	.16	.10	.55
BNC-PGH	High-level	.52	.53	.52	.51	.92
BNC-PGH	All	.61	.52	.56	.56	.75
MapTask	(baseline)	.11	1.0	.20	.11	.50
MapTask	TeaSimSum	.29	.03	.06	.12	.51
MapTask	High-level	.38	.38	.38	.34	.81
MapTask	All	.41	.63	.50	.55	.76
Switchboard	(baseline)	.002	1.0	.005	.002	.50
Switchboard	OCRProportion	0	0	0	0	.50
Switchboard	High-level	.54	.52	.53	.50	.98
Switchboard	All	.52	.60	.56	.58	.80

Table 2

NTRI detection: Precision, Recall, F-score and Area Under Curve (AUC) metrics for NTRI utterances, using 10-fold cross-validation. We show AUC for the precision-recall curve for NTRIs (AUC-PRC) as well as the more usual receiver-operator curve (AUC-ROC); AUC-PRC is more informative with unbalanced data.

Corpus	Features	P	R	F	Correl.	F rm
PCC all	words	.648	.555	.598	.798**	-
PCC all	words+POS	.660	.585	.620	.805**	-
BNC-CH	words	.350	.446	.392	.530**	-
BNC-CH	words+POS	.397	.438	.417	.583**	-
Switchboard	words	.910	.758	.827	.962**	.749
Switchboard	words+POS	.928	.785	.850	.956**	.781

Table 3

Self-repair detection: STIR's per-utterance performance on our corpora in terms of rp_{start} *(repair onset) detection and the Spearman's rank correlation between STIR and the annotators' repair rates (*rp_{start} *per utterance) per speaker (**=p<0.001). The reparandum word detection accuracy is also given for Switchboard.*

System (evaluation)	F rm (word)	F rp_{start}	Correl.
RNN (+ partial)	0.631	0.751	0.948**
RNN (- partial)	**0.668**	**0.790**	**0.956****
STIR +POS (+ partial)	**0.781**	**0.850**	**0.956****
STIR +POS (- partial)	0.768	0.833	0.937**

Table 4

The effect of partial words: Comparison of STIR's performance to an RNN disfluency tagger testing on Switchboard heldout data with and without partial words. STIR improves whilst the RNN suffers with partial words.

(a)

Reparandum + Interregnum length	(support)	RNN	STIR (-POS)	STIR (+POS)
1	(1254)	.756	.852	.874
2	(531)	.590	.730	.782
3	(227)	.397	.600	.688
4	(106)	.286	.533	.559
5	(50)	.098	.370	.430
6	(25)	.000	.308	.500
7	(11)	.000	.000	.154
8	(6)	.000	.250	.286

(b)

Repair Type	(support)	RNN	STIR (-POS)	STIR (+POS)
repetition	(1022)	.923	.970	.969
substitution	(1061)	.536	.708	.759
delete	(132)	.366	.453	.407

Table 5

Self-repair detection error analysis: (a) F-score for detecting the correct repair start word and reparandum start word of repairs with different combined reparandum and interregnum lengths; (b) F-score for detecting repair onset word of different types. Compared with off-the-shelf RNN disfluency tagger on the SWBD held-out data.

Rank	BNC-PGH	SWBD	MAPTASK
1	qMarkProportion	qMarkProportion	BackChProportion
2	qMarks	qMarks	BackChWholeTurn
3	WhProportion	WhProportion	NumWords
4	WhWords	OCRProportion	TeaSimSum
5	OCRProportion	OpenClassRepair	RepeatedProportion
6	NumWords	SelfW2vSimSum	RepeatedPos
7	OpenClassRepair	NumWords	RepeatedWords
8	RepeatedProportion	BackChProportion	W2vSimSum
9	SelfW2vSimSum	NumBackchannel	TeaSimMax
10	NumBackchannel	TeaSimMean	W2vSimMax
11	SelfTeaSimSum	W2vSimMean	RepeatedPosProportion
12	BackChProportion	W2vSimSum	RepeatedSelfPosProportion
13	SelfTeaSimMax	RepeatedSelfPos	W2vSimMin
14	SelfW2vSimMax	TeaSimMax	TeaSimMin
15	RepeatedPosProportion	W2vSimMax	NumBackchannel
16	RepeatedSelfPos	SelfTeaSimMax	W2vSimMean
17	RepeatedWords	SelfW2vSimMax	TeaSimMean
18	RepeatedPos	RepeatedSelfWords	RepeatedSelfPos
19	SelfW2vSimMin	WhWords	SelfTeaSimSum
20	SelfTeaSimMin	RepeatedPosProportion	SelfW2vSimSum
21	BackChWholeTurn	RepeatedProportion	RepeatedSelfProportion
22	TeaSimSum	RepeatedSelfProportion	SelfTeaSimMax
23	W2vSimMean	SelfTeaSimSum	SelfW2vSimMax
24	RepeatedSelfWords	SelfW2vSimMin	SelfTeaSimMin
25	TeaSimMean	SelfTeaSimMin	SelfW2vSimMin

Table 6

Feature ranker (Information Gain) for other-repair (NTRI) detection: top 15 features in order, using 10-fold cross-validation on BNC-PGH, SWBD and MAPTASK datasets. Note that question marks (qMarks, qMarkProportion) are not transcribed in MAPTASK.

average merit	average rank	attribute
0.139 (+- 0.002)	1 (+- 0.00)	H^{pos}
0.131 (+- 0.001)	2 (+- 0.00)	WML^{pos}
0.126 (+- 0.001)	3.4 (+- 0.66)	WML^{lex}
0.125 (+- 0.003)	4 (+- 1.10)	s^{pos}
0.122 (+- 0.001)	5.9 (+- 0.94)	$w_{i-1} = w_i$
0.122 (+- 0.001)	5.9 (+- 0.70)	BestWMLboost^{lex}
0.122 (+- 0.002)	5.9 (+- 1.22)	InformationGain^{pos}
0.119 (+- 0.001)	7.9 (+- 0.30)	BestWMLboost^{pos}
0.098 (+- 0.002)	9 (+- 0.00)	H^{lex}
0.08 (+- 0.001)	10.4 (+- 0.49)	ΔWML^{pos}
0.08 (+- 0.003)	10.6 (+- 0.49)	ΔH^{pos}
0.072 (+- 0.001)	12 (+- 0.00)	$POS_{i-1} = POS_i$
0.066 (+- 0.003)	13.1 (+- 0.30)	s^{lex}
0.059 (+- 0.000)	14.2 (+- 0.40)	ΔWML^{lex}
0.058 (+- 0.005)	14.7 (+- 0.64)	BestEntropyReduce^{pos}
0.049 (+- 0.001)	16.3 (+- 0.46)	InformationGain^{lex}
0.047 (+- 0.004)	16.7 (+- 0.46)	BestEntropyReduce^{lex}
0.035 (+- 0.004)	18 (+- 0.00)	ΔH^{lex}
0.024 (+- 0.000)	19 (+- 0.00)	$w_{i-2} = w_i$
0.013 (+- 0.000)	20 (+- 0.00)	$POS_{i-2} = POS_i$
0.01 (+- 0.000)	21 (+- 0.00)	$w_{i-3} = w_i$
0.009 (+- 0.000)	22 (+- 0.00)	edit
0.006 (+- 0.000)	23 (+- 0.00)	$POS_{i-3} = POS_i$

Table 7

Feature ranker (Information Gain) for rp_{start} detection- 10-fold x-validation on Switchboard heldout data.

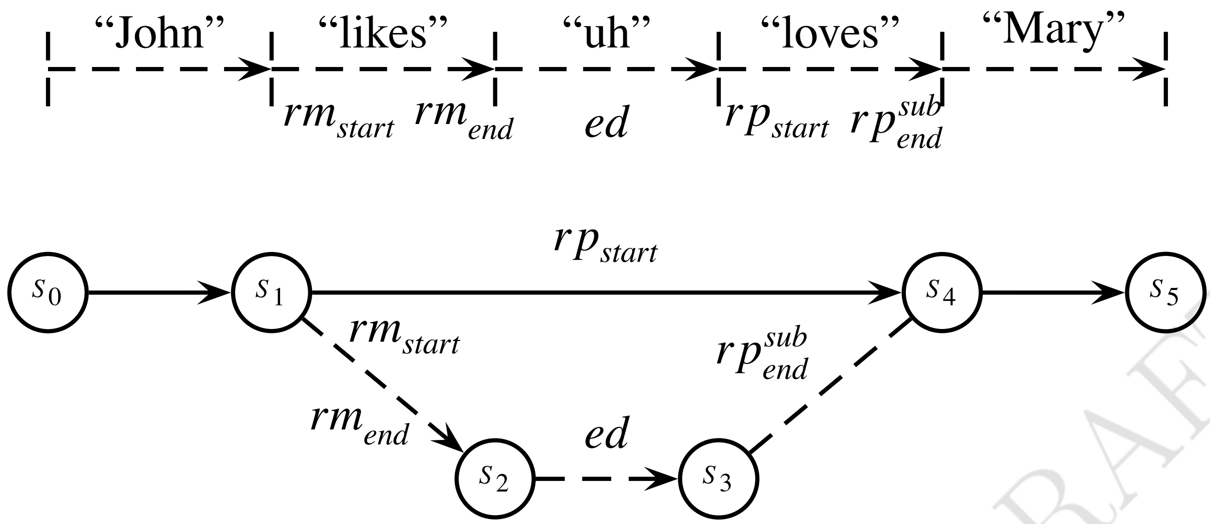

Figure 1. STrongly Incremental Repair detection (STIR); application to the utterance "John likes, uh, loves Mary", with incoming words and STIR's output tags at top.

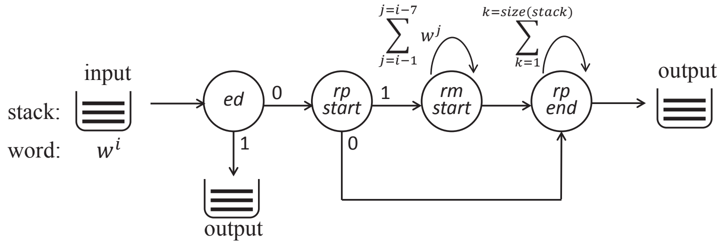

Figure 2. STIR's pipeline of classifiers

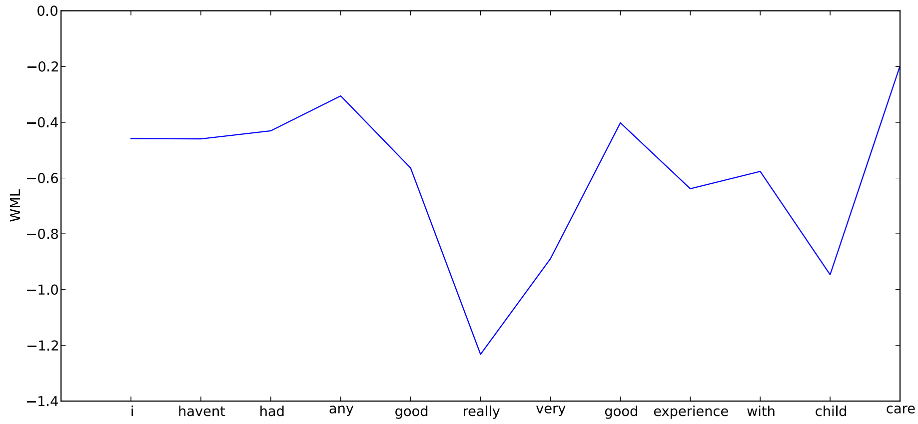

Figure 3. WML^{lex} values for trigrams for a repaired utterance exhibiting the drop at the repair onset