
Issues for Processing Speech

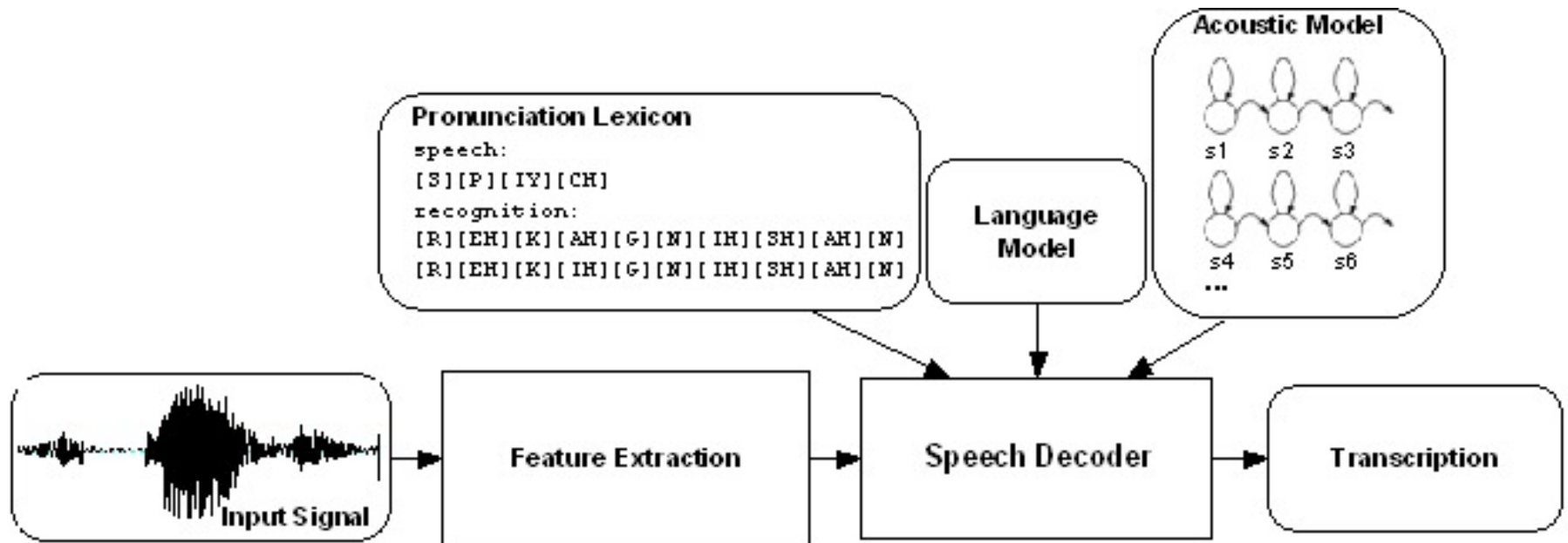
Mary Harper

Spontaneous Speech Challenges Language Processing Approaches



so we need but how do we get them out I say we have we set a string of charges that will root them out the back so t- the charges start at the front and just explode and blow a little something up but are really really loud and and marsupials have really good ears so that'll be real that'll really frighten them

Speech Recognition



The Challenges of Spontaneous Speech for Language Processing

- Difficult for the recognizer
 - ASR errors (insertions, deletions, and substitutions)
 - Phenomena atypical of textual sources (e.g., filled pauses, speech repairs)
 - Acoustic challenges (fragments, filled pauses, coarticulation)
 - Language models do not currently model disfluencies adequately
- Recognition output is difficult for humans to read
- Recognition output is difficult for NLP
 - Sentence boundaries are NOT provided and ASR segments are often inappropriate
 - Utterances are different (planned on the fly) from written text
 - Much of spoken language is used for organizing the communication (e.g., “And so”).
 - Speech repairs are challenging.

Assorted Spontaneous Speech Phenomena

- Filled pauses:

*I think it's **uh** refreshing to see the **uh** support . . .*

- Parentheticals:

*but **you know** I was reading the other day . . .*

- Speech repairs:

***why didn't he** why didn't she stay at home*

- Partial Words:

*cut between these **t-** these trees*

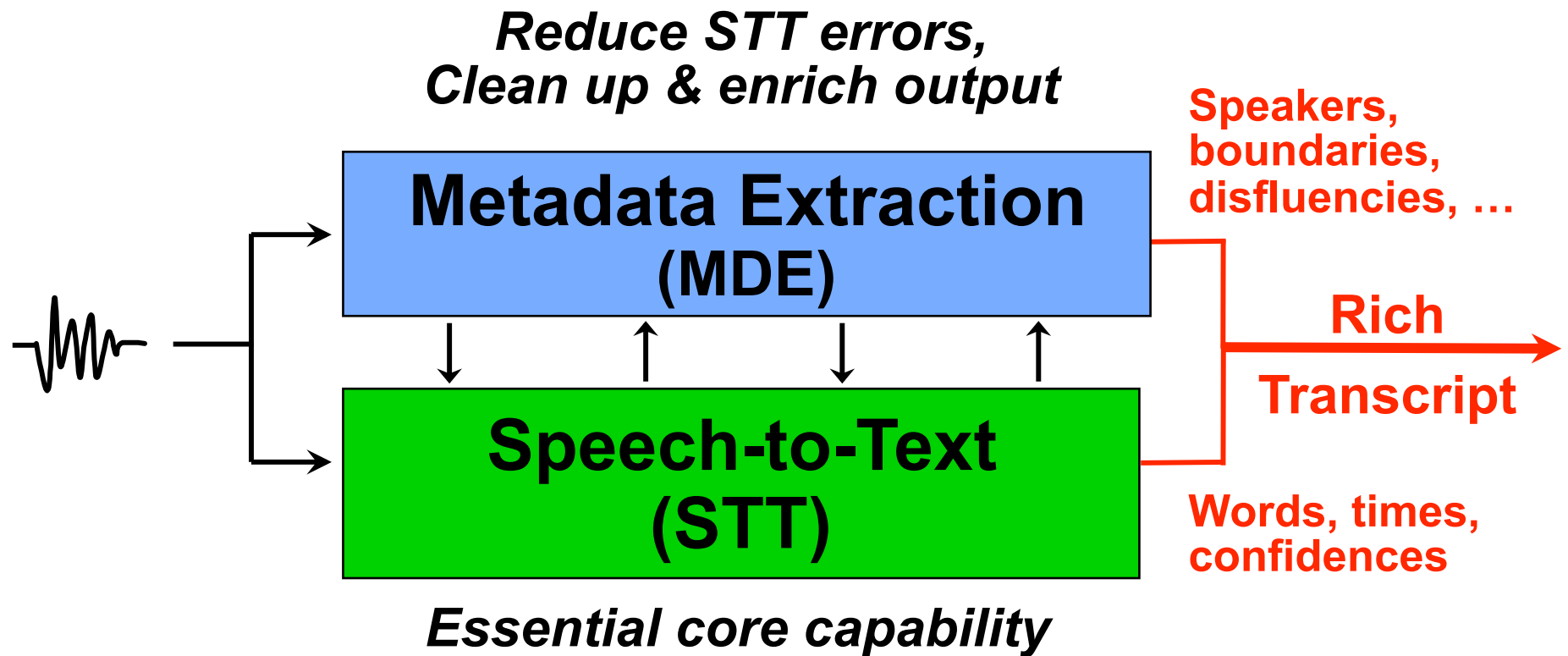
- "Ungrammatical" constructions:

my friends is visiting me

Enrich Word Stream with Structural Metadata

- **[so we need]** * but how do we get them out /?
 - **<I say> [we have]** * we set a string of charges that will root them out the back /.
 - **<so> [t-]** * the charges start at the front and just explode and blow a little something up but are really really loud /.
 - **[and]** * and marsupials have really good ears /.
 - **<so> [that'll be real]** * that'll really frighten them /.
-

Metadata Extraction and Transcription



Structural Metadata Extraction Tasks

- **Sentence Unit (SU) detection:** find the sentence-like units and their subtypes
 - **Filler word detection:** filled pauses, discourse markers (e.g., <you know>), explicit editing terms
 - **Interruption point (IP) detection** (e.g., we have * we set a string of charges)
 - **Edit word detection:** reparandum region of a speech repair (e.g., [we have] * we set a string of charges)
-

Motivation for Rich Transcriptions

- Adding additional information to a transcription should:
 - **Aid downstream language processing (provide sentence boundaries, indicate structure of disfluencies)**
 - Improve readability to humans (adding punctuation, removing disfluencies) [e.g., MITLL readability experiments]
 - Improve ASR performance (e.g., feedback metadata information to recognizer to aid language models) [e.g., Work by Sebastien Coquoz, visiting ICSI from EPFL]

Feedback Structural Information to ASR

[Work by Sebastien Coquoz, visiting ICSI from EPFL]

- Motivation:
 - Linguistic segments are more appropriate for LMs than acoustically segmented (speech vs. non-speech) chunks
 - Error analysis of BN recognition reveals a higher error rate at ASR segment boundaries
- Approach: use automatically-detected sentence boundaries to re-segment speech and then re-recognize
- Results on BN corpus RT-03 eval set:
 - Recognizer automatic ASR segments: 14.0% WER
 - Use reference boundary information: 13.0%
 - Using system boundary information: 13.3% so far
 - This segmentation helps STT!

The Challenge of Parsing Speech

- There is a mismatch between ASR systems and statistical parsers:
 - Segments processed by an ASR system do not typically correspond to segments that statistical parsers normally work with.
 - ASR systems:
 - Produce long word strings without punctuation,
 - Word strings often contain errors (insertions, deletions, and substitutions),
 - Word strings contain phenomena that do not typically occur in textual sources (e.g., filled pauses, speech repairs).
 - Traditional parsers are text-based:
 - Don't use acoustic cues,
 - Process sentences not segments,
 - Process input without word errors,
 - Process textual input without spontaneous speech phenomena.

How to Enable Effective Downstream Processing of Speech

- **Metadata extraction**
 - Providing sentence boundaries and disfluency annotations
 - Challenging: speech is difficult
 - **Parsing**
 - Structure enables other downstream processing
 - Challenging: parsing has been traditionally text-centered
 - Need to deal with speech related phenomena
 - Performance metrics exist for parsing text that need to be adapted to speech
-

Data Resources

- The RT'04 conversational telephone speech data, annotated with structural metadata, was used in the RT'04 MDE benchmark tests.
- Gold standard parses from the LDC treebanking team for dev, dev2, and eval sets.
- Recognition output from state-of-the-art recognizers for the EARS RT'04 data.
- Using this new data allowed us to evaluate the synergy between parsing and MDE system performance.

	conversations	# SUs	# words
dev	72	11K	71K
dev2	36	5K	35K
eval	36	5K	34K

Resource

JHU Speech Parsing Corpus (LDC2005E15):

A unified conversational speech resource with consistent metadata markups and parse trees

- Metadata markup
 - Sentence boundaries
 - Speech disfluencies
 - Treebank parse trees
 - Syntactic structure
 - Restarts
-

An Example

<FL_ST> well <FL_END> <EDIT_ST> i- <EDIT_END> i
know , it's cold outside there now , huh

```
(S1 (S (INTJ (UH well))
  (EDITED (S (NP (PRP i-))) (DISFL-IP +))
  (NP (PRP i))
  (VP (VBP know)
    (, ,)
    (SBAR (S (NP (PRP it))
      (VP (BES 's)
        (ADJP (JJ cold))
        (ADVP (RB outside))
        (ADVP (RB there)
          (RB now))))
      (, ,)
      (INTJ (UH huh))))
    (. ?)))
```

An Example

<FL_ST> well <FL_END> <EDIT_ST> i- <EDIT_END> i
know it's cold outside there now huh

```
(S1 (S (INTJ (UH well))
  (EDITED (S (NP (PRP i-))))
  (NP (PRP i))
  (VP (VBP know)
    (SBAR (S (NP (PRP it))
      (VP (BES 's)
        (ADJP (JJ cold))
        (ADVP (RB outside))
        (ADVP (RB there)
          (RB now))))
      (INTJ (UH huh))))))
```

Data

- 144 conversations, 140,000 words, 21,000 SUs (syntactic/semantic units)
 - Transcribed English conversational telephone speech originally developed for the DARPA EARS (Efficient, Affordable, Reusable Speech-To-Text) Program
 - Switchboard (LDC97S62) and Fisher Protocol (LDC2004E16, LDC2004E29, LDC2005E73)
 - The Fisher data was carefully transcribed at LDC using RT-04 Transcription Specification, Version 3.1
-

Measuring Parse Accuracy on Speech

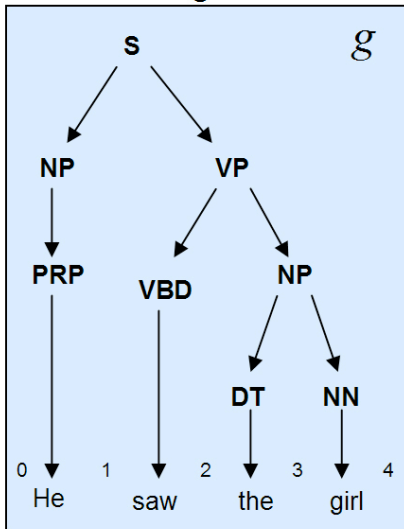
- Parsing techniques are now being applied to automatic speech recognition (ASR) output with
 - Automatic transcripts
 - Automatically generated sentence segments (SUs)that differ in many cases from the gold words and segments.
 - This creates the need to develop and evaluate new methods for determining spoken parse accuracy that support evaluation when the yields of gold-standard parse trees may differ from parser output due to both:
 - Transcription differences (wrong words)
 - Sentence segmentation differences (wrong boundaries)
-

Parsing Metrics: Brackets

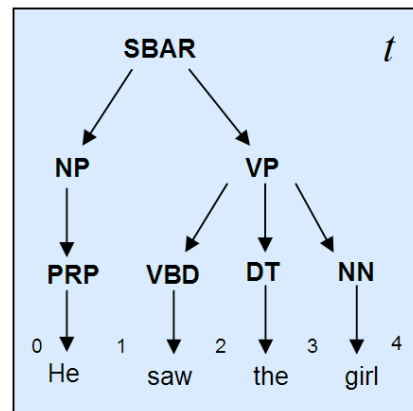
$brackets(g) = \{S(0,4), \underline{NP(0,1)}, \underline{VP(1,4)}, NR(2,4)\}$

$brackets(t) = \{SBAR(0,4), \underline{NP(0,1)}, \underline{VP(1,4)}\}$

gold standard



evaluation tree



$$LP(t, g) = \frac{2}{3} = 66.66\%$$

$$LR(t, g) = \frac{2}{4} = 50.00\%$$

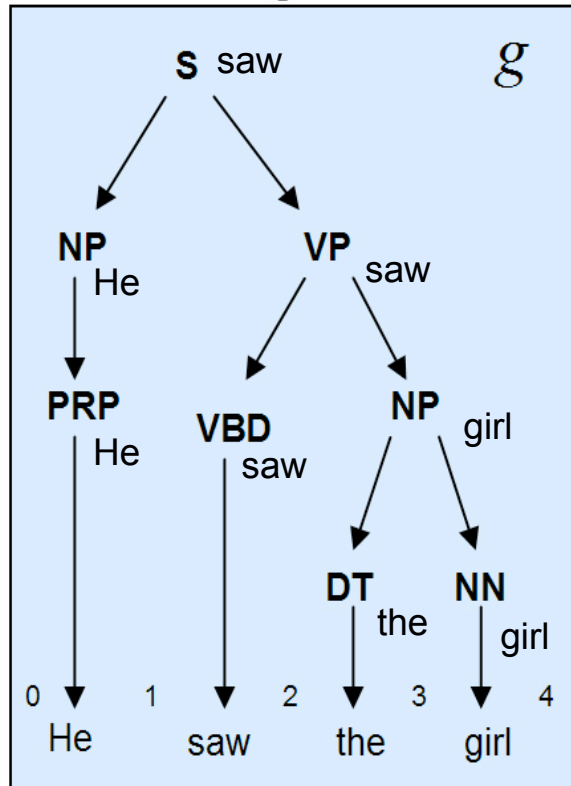
$$F_{meas}(t, g) = \frac{2 \cdot 66.66 \cdot 50}{66.66 + 50} = 57.14\%$$

State of the art on WSJ PTB is 91% F-measure with reranking parser.

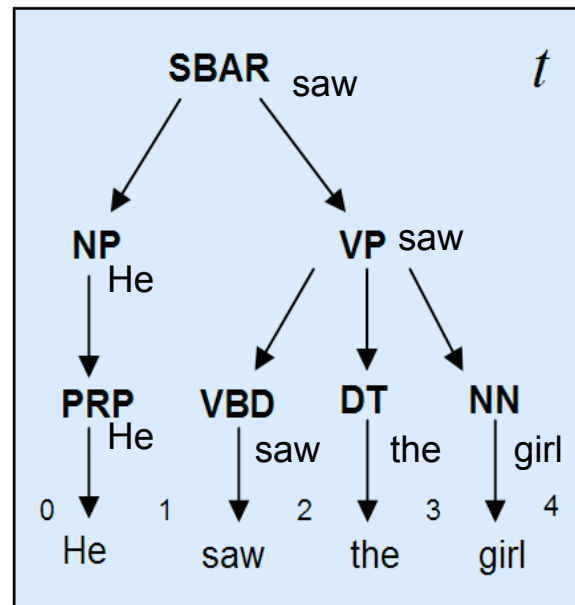
Parsing Metrics: Head Dependency

Dep(g) = {(saw S/NP He) (saw VP/NP girl)
(girl NP/DT the) (saw S/TOP)}
Dep(t) = {(saw SBAR/NP He) (saw VP/NN girl)
(saw VP/DT the) (saw SBAR/TOP)}

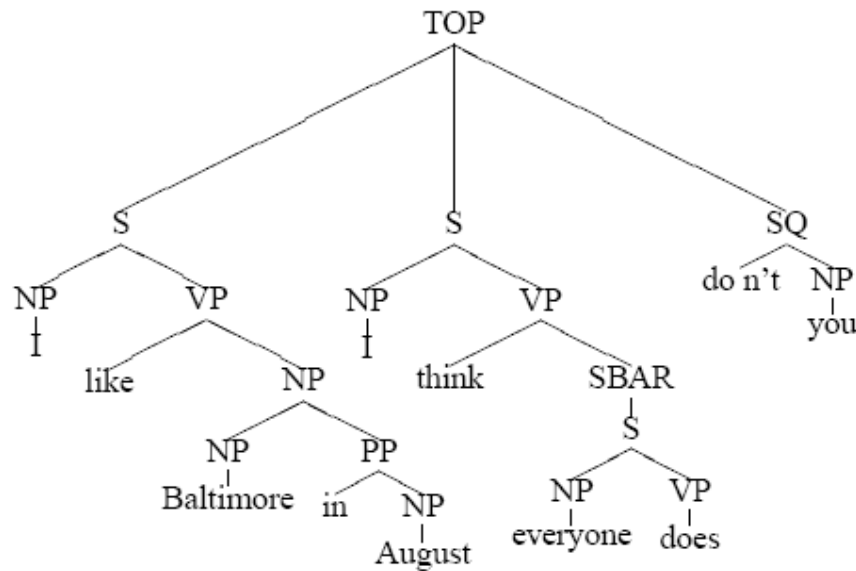
gold standard



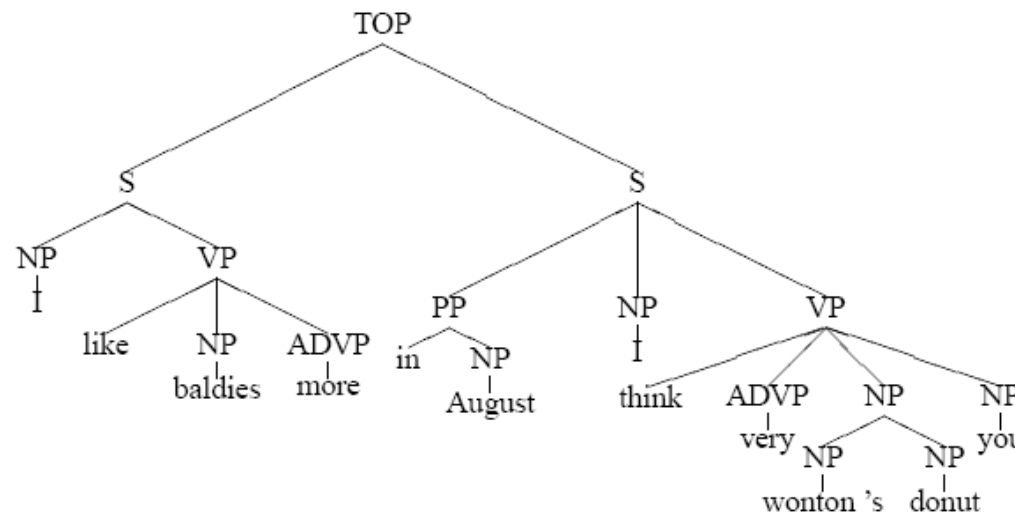
evaluation tree



Issues for Gold and Test Match



(TOP/S,like)
 (like,S/NP,I)
 (like,VP/NP,Baltimore)
 (Baltimore,NP/PP,in)
 (in,PP/NP,August)
 (TOP/S,think)
 (think,S/NP,I)
 (think,VP/SBAR,does)
 (does,S/NP,everyone)
 (TOP/SQ,do)
 (do,SQ/RB,n't)
 (do,SQ/NP,you)



(TOP/S,like)
 (like,S/NP,I)
 (like,VP/NP,baldies)
 (like,VP/ADVP,more)
 (TOP/S,think)
 (think,VP/PP,in)
 (in,PP/NP,August)
 (think,S/NP,I)
 (think,VP/ADVP,very)
 (think,VP/NP,donut)
 (donut,NP/NP,'s)
 ('s,NP/NP,wonton)
 (think,S/NP,you)

Measuring Parse Accuracy on Speech

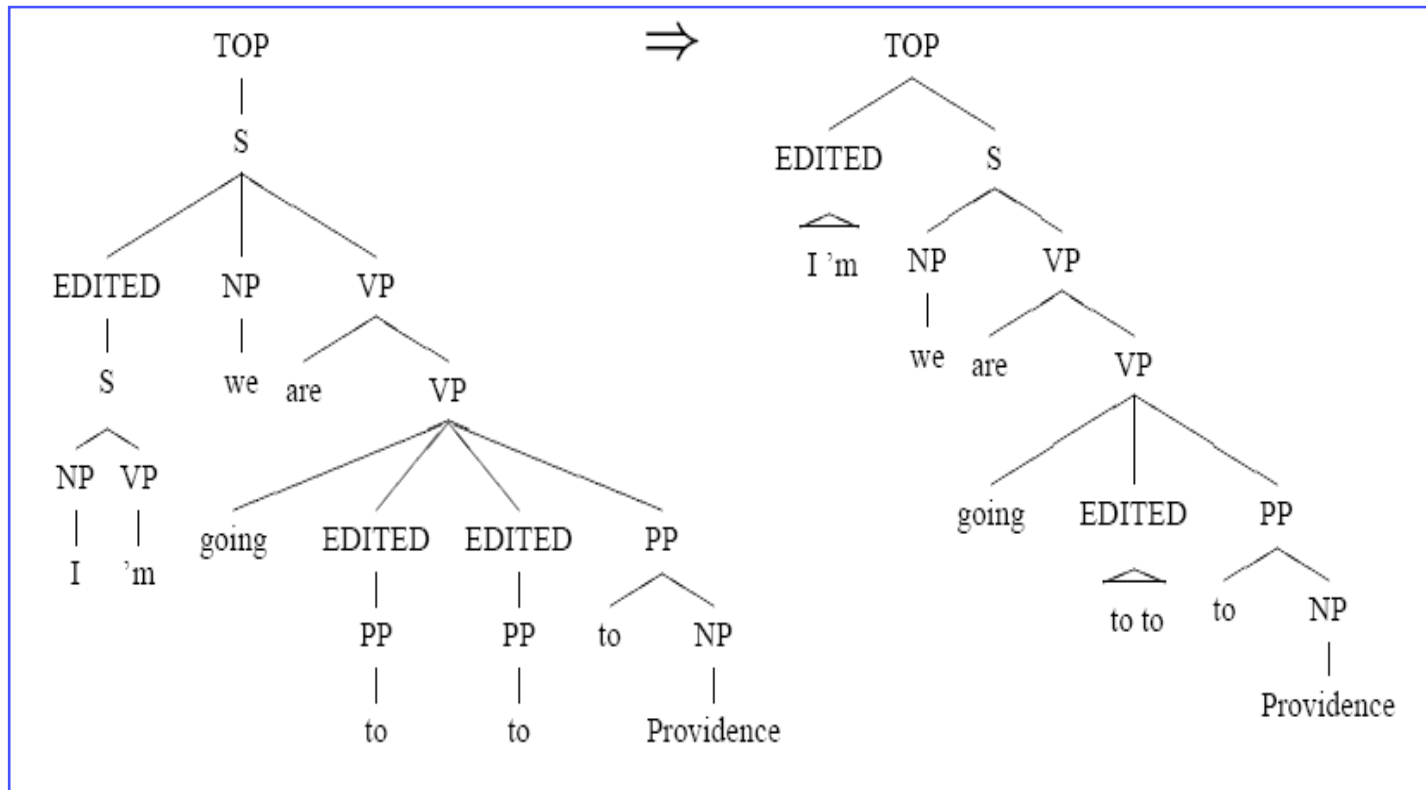
- How do we measure parsing accuracy given:
 - Word mismatch
 - SU mismatch
- Alignment:
 - Reference transcript and ASR output can be aligned
- Metrics investigated:
 - bracket-based (i.e., adapt Parseval metrics)
 - dependency-based



Different Words and SUs in Gold and Test Parses

I like Baltimore	in August		I think everyone does		do n't you
I like baldies more		in August	I think very wonton 's		donut you
I	I		000		
like	like		000		
Baltimore	baldies		001	← substitution	
	more		010		
in	in		000		
August	August		000		
I	I		000		
think	think		000		
everyone	very		001		
does	wonton		001		
	's		010	← insertion	
do	donut		001		
n't			100	← deletion	
you	you		000		

Handling EDITED Scoring



- Remove internal structure of EDITED nodes
- Merge adjacent EDITED nodes
- Ignore (like punctuation) for deciding other constituents' span

The Scoring Tool

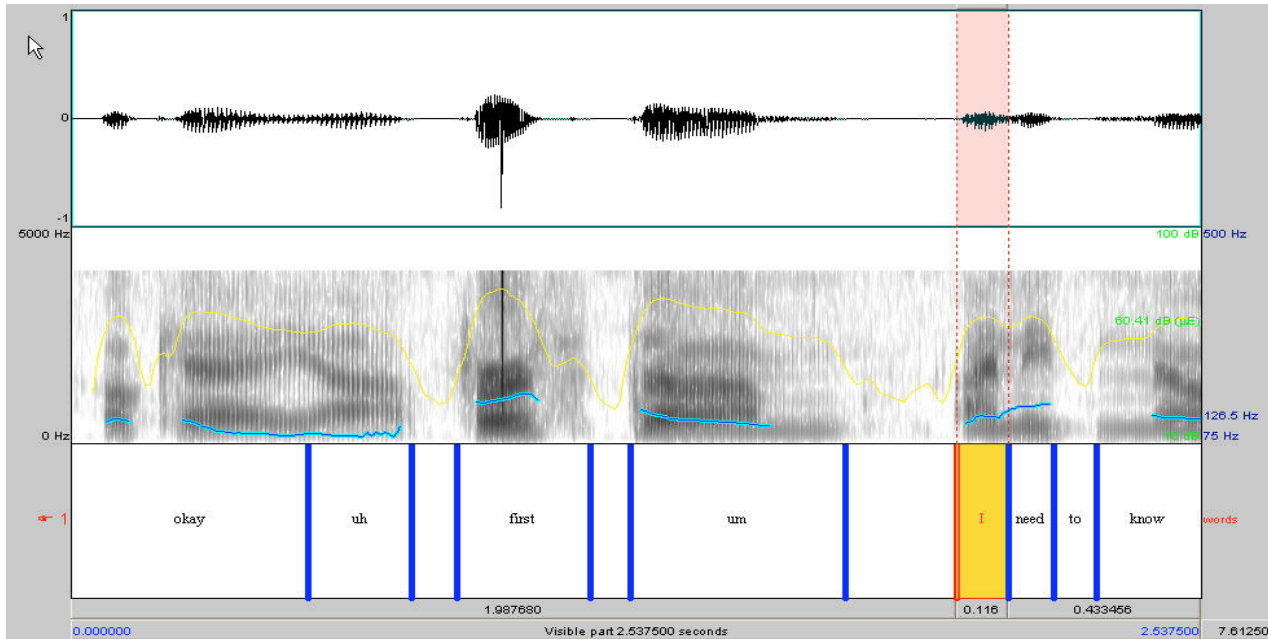
- The SParseval tool, implemented in C, was designed to support scoring at the level of a demarcated chunk of speech such as a conversation side, as well as to support more traditional text-based scoring methods.
- SParseval, invoked on the command line in Unix, was designed to be flexibly configurable to support a wide variety of scoring options.

```
Usage: sparseval [-opts] goldfile parsefile
```

```
Options:
```

```
-p file      evaluation parameter file
-h file      head percolation file
-a file      string alignment file
-F file      output file
-l           goldfile and parsefile are lists
              of files to evaluate
-b           no alignment
              (bag of head dependencies)
-c           conversation side
-u           unlabeled evaluation
-v           verbose
-z           show info
-?           info/options
```

MDE for Spontaneous Speech



Gold Standard (Human) Transcription:

okay uh first um I need to know uh ho- how do you feel about uh about sending um an elderly uh family member to a nursing home

Knowledge Sources: Textual Information

- Reference transcripts are annotated with structural metadata, e.g.,
**<FL_ST> well <FL_END> <EDIT_ST> i- <EDIT_END> i know it's
cold outside there now huh I?**
- Word n-grams
- Other textual information, at different levels of granularity, and from additional corpora
 - Capture additional information
 - Address sparse data problem, improve generalization
 - Investigated:
 - Automatically induced classes (data-driven)
 - Part-of-speech tags (linguistic)
 - Syntactic chunk tags
 - Additional un-annotated corpora for word-based LMs

Knowledge Sources: Prosody

- Not just “what is said”, “who said it”, but *“how it is said”*
- Prosody conveys information about
 - syntactic structure and phrasing
“Don’t turn right.” vs. “Don’t turn, right?”
 - Semantic, discourse, and emotion
“Yeah.” vs. “yeah?” vs. “Yeah!”
- Prosody is important for metadata detection
 - Provides complementary information, combines well with textual information
 - Independent of lexical information, may be more robust in face of word errors

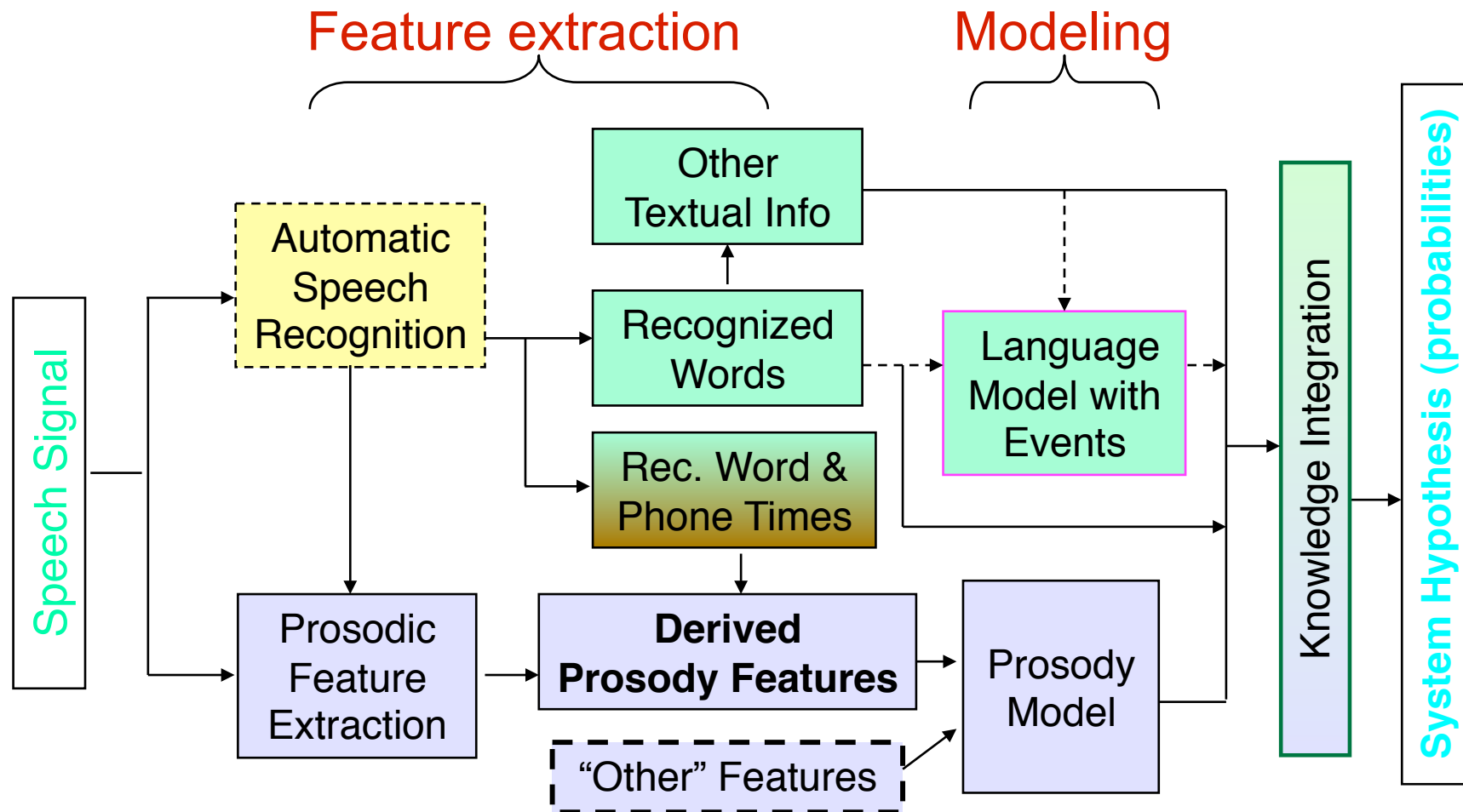
Prosodic Features

- Features are based on longer time range than recognition units
- Earlier work used hand labeled prosodic features. They are automatically extracted in these experiments
- Features are associated with each inter-word boundary:
 - Duration, pause, and speaking rate (with normalization)
 - F0 features (stylized values, slope patterns, speaker normalization)
 - Energy features

Prosody Model

- Use a classifier to calculate $P(E_i | F_i)$
 - E : metadata type, F : prosodic features
- Chose to use a decision tree classifier (rather than SVM, GMM, etc.)
- Desired properties of a classifier:
 - Provide posterior probability estimates
 - Handle missing features, categorical features
 - Handle large data set
 - No data warping/scaling required
 - Easy interpretation of the useful features

General Modeling Framework for Metadata



Integration Approaches

	HMM	Maxent	CRF
Discriminatively trained	N	Y	Y
Easy to handle correlated features	N	Y	Y
Models sequential information	Y	N	Y
Training is computationally efficient	Y	N	N

Features in Maxent and CRF

- Word N-grams
 - Part-of-speech N-grams
 - N-grams of automatically-induced class
 - Cumulative binned posterior probabilities from the prosody model
 - Cumulative binned posterior probabilities from the additional language models
-

MDE Scoring Metric

- MDE scoring tool first aligns recognition words to reference transcripts and then maps metadata events
 - Error rate = # errors / # **reference events**
- An example of SU detection output:

Reference: w w | w w w w |

System: w w w | w w w |

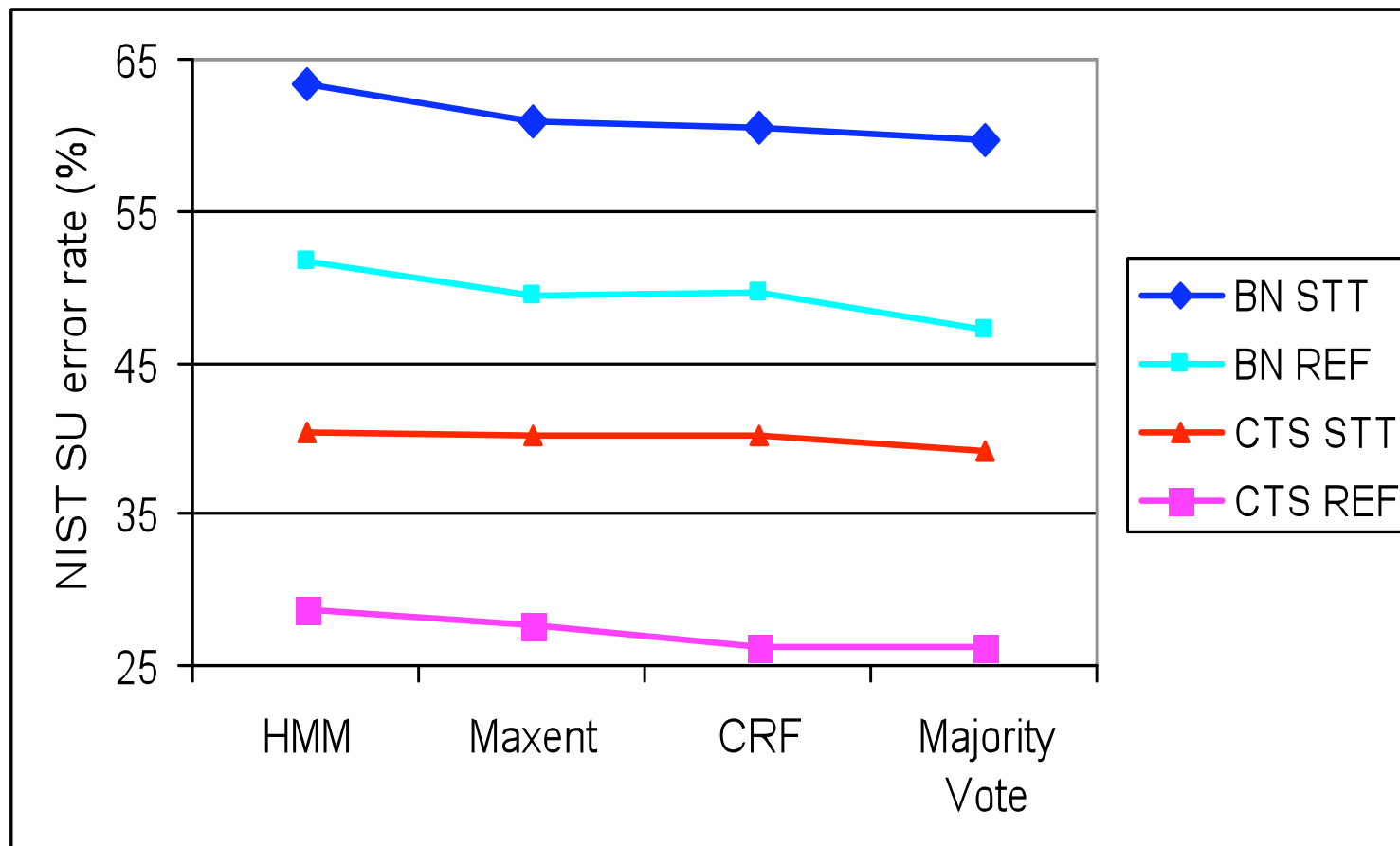
del ins correct

NIST error (per SU event) = 2/2 = 100%

(NIST metric tends to be high)

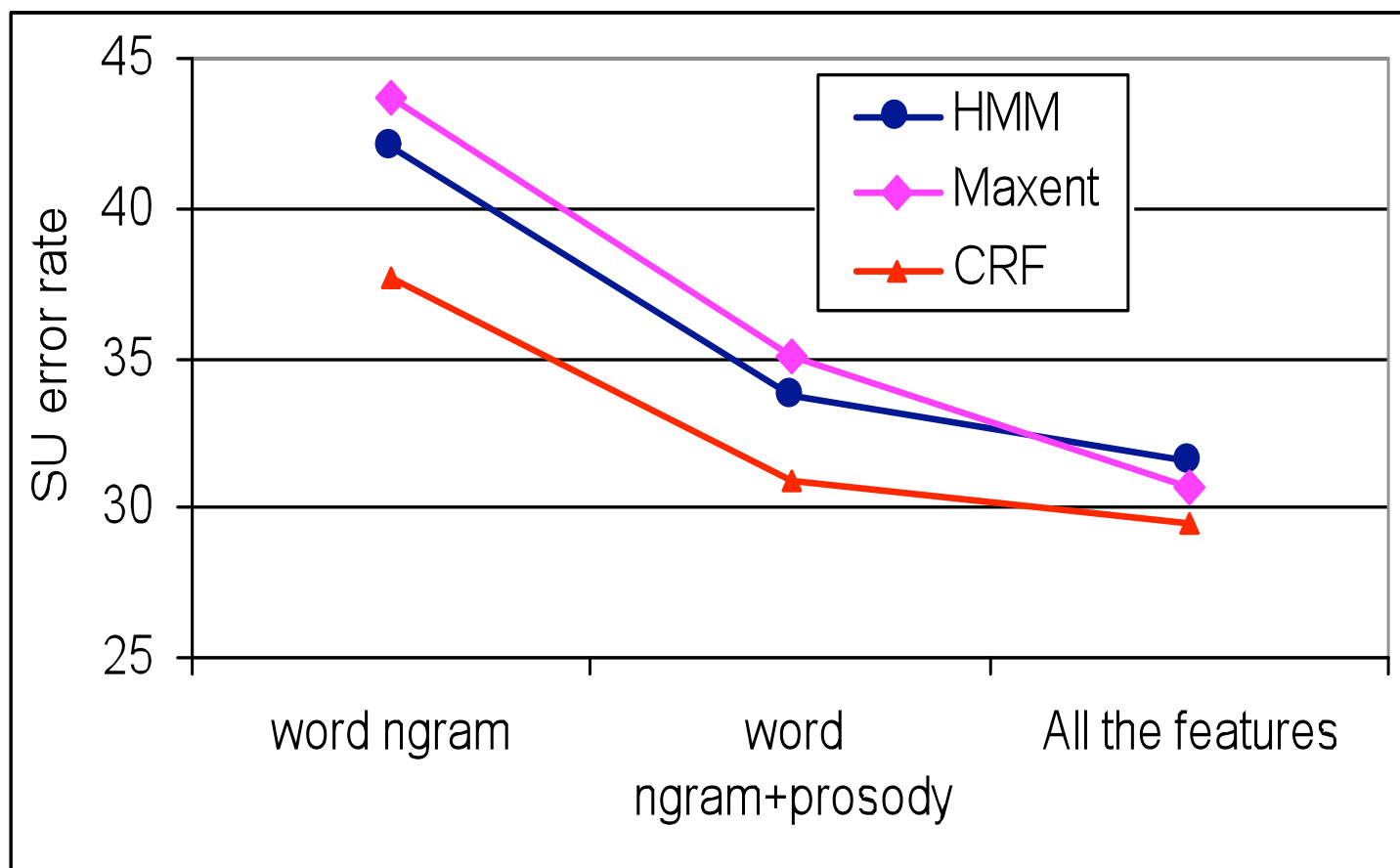
Per-boundary-based error = 2/6 = 33%

SU Detection over Approaches



Note: STT WERs = 14.9% CTS, 11.7% BN

Impact of Different Knowledge Sources



Findings for SU Detection

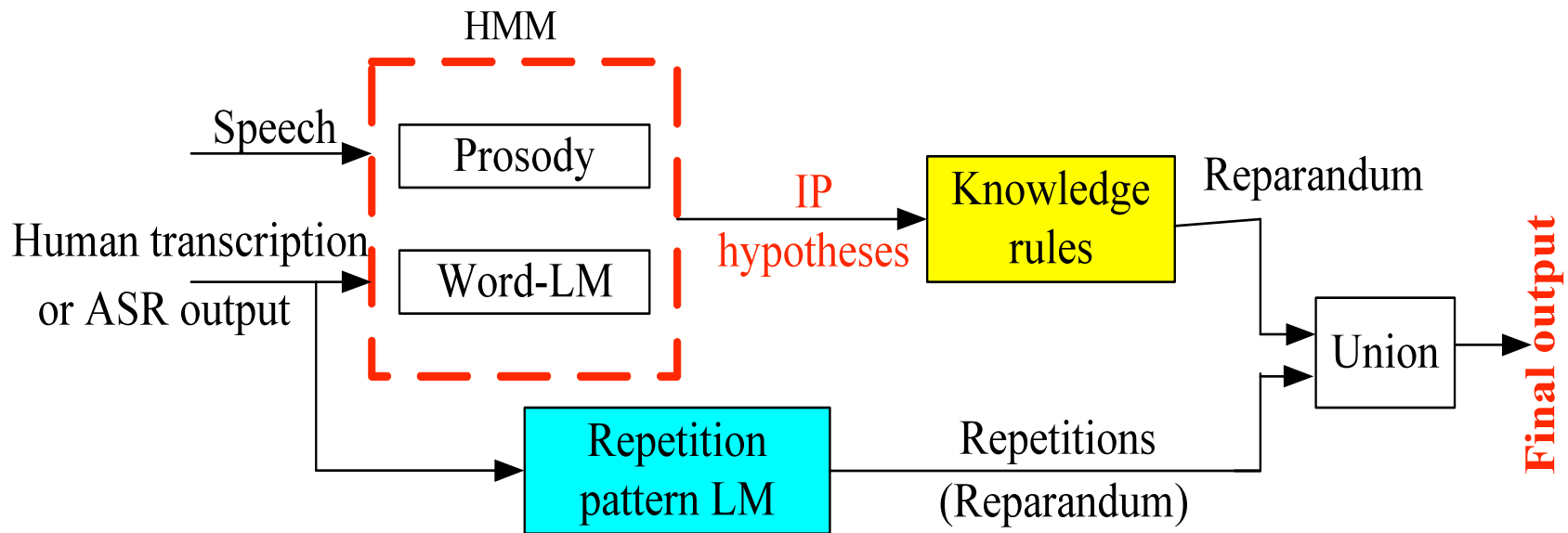
- Combination of multiple knowledge sources achieves best performance
- There are key differences between CTS and BN, e.g.
 - **higher STT error rate for CTS**
 - **fewer events per word in BN (hence smaller denominator)**
 - **larger vocabulary, more data sparseness for BN**
 - **prosody contributes more on BN than CTS**
- Performance degrades using recognition outputs, with greater impact on the LMs
- Maxent/CRF are better at combining textual features, but the current modeling approaches could better exploit prosodic information

SU and Parsing

- It would be possible to perform supervised training on parsers so that the parser identifies not only the underlying structure of an SU but the underlying structure of a entire conversation side.
- This approach is infeasible due to issues of computational complexity, not to mention memory issues.
- Fortunately, it is a simple matter to pass n-best SU hypotheses from a structural metadata system to the parser.

Edit Word Detection (I)

- System diagram (using HMM for IP detection)

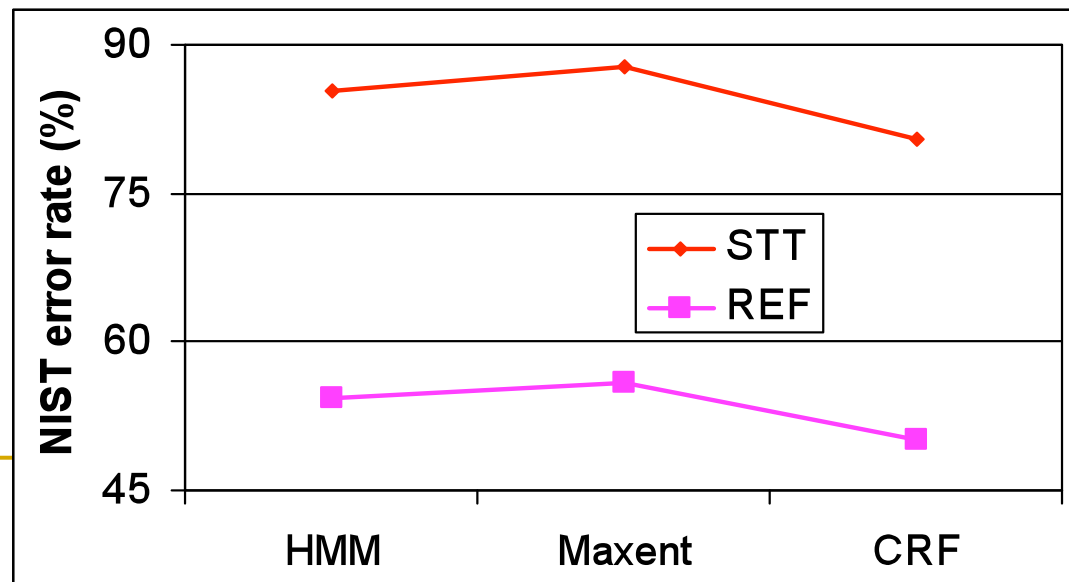


- Maxent can also be used in place of HMM box for IP detection

Edit Word Detection (II): CRF Jointly Detects Edit Region and IP

I	B-E + IP	begin edit & IP
I	I-E	inside edit
work	I-E + IP	inside edit & IP
I'm	O	outside edit
a	O	outside edit
researcher	O	outside edit

CTS results



Findings for Edit Detection

- Performance degrades a lot in STT condition, due to word errors and lack of word fragment information
- CRF achieves a lower error rate than HMM
- Prosody model less helpful than for SU task
- Observed differences across CTS and BN
 - disfluencies are rarer in BN, more repeats
 - reference condition relatively easier for BN
 - complex disfluency structures in CTS, harder task
- Syntactic information would be helpful

Edited Regions and Parsing

- It would be possible to perform supervised training on parsers so that the parser identifies Edited regions within an SU.
- Although supervision can be provided via an appropriately annotated treebank and is computationally tractable, there are issues of representation that must be understood.
- It is easy to pass n-best MDE hypotheses from a structural metadata system to the parser.

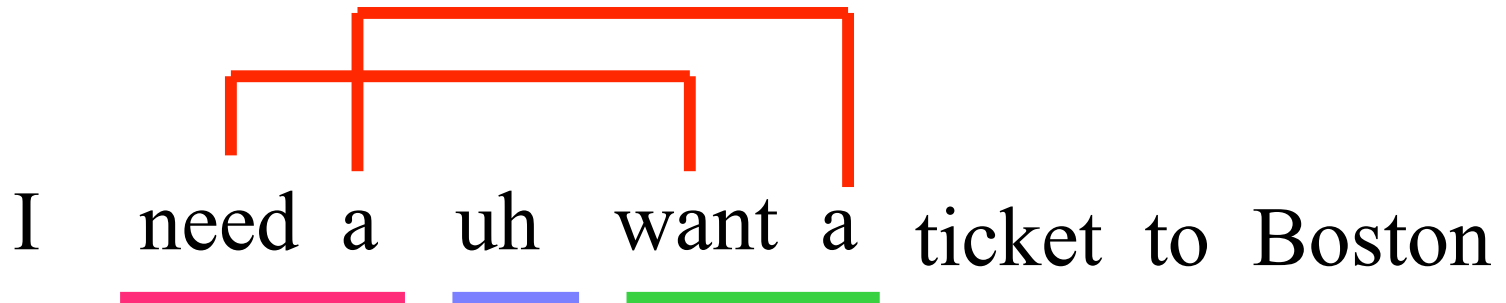
EDITED Constituents in Treebank

*<FL_ST> well <FL_END> <EDIT_ST> i- <EDIT_END> i know ,
it's cold outside there now , huh*

(S1 (S (INTJ (UH well))
 (EDITED (S (NP (PRP i-)))
 (DISFL-IP +))
 (NP (PRP i))
 (VP (VBP know)
 (, ,)
 (SBAR (S (NP (PRP it))
 (VP (BES 's)
 (ADJP (JJ cold))
 (ADVP (RB outside))
 (ADVP (RB there)
 (RB now))))
 (, ,)
 (INTJ (UH huh))))))
(. ?)))

Speech Repairs

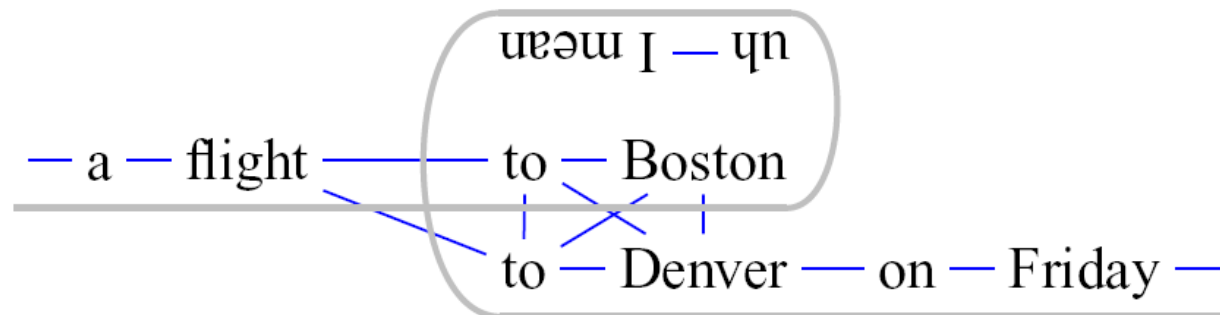
- Repetitions and Content Replacements speech repairs have a structure that involves cross-serial dependencies.
- Many parsers are unable to model the cross-serial dependencies explicitly (exceptions are TAG and CDG)



Reparandum Editing Term Correction

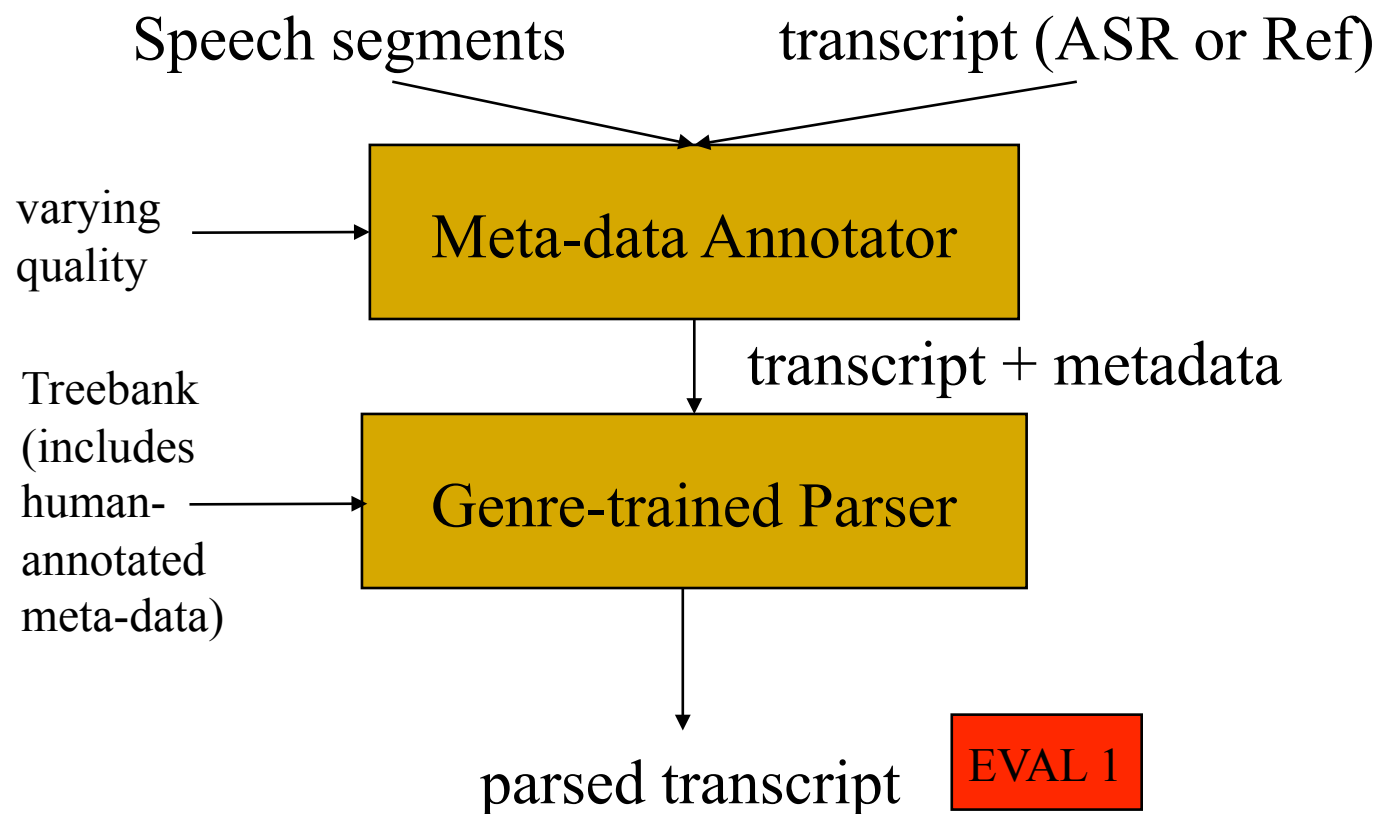
Repairs and Parsing (given SU)

Charniak & Johnson. *Edit Detection and Parsing for Transcribed Speech*. NAACL 2001.



- *Correction* often a “rough copy” of *reparandum*, involving crossed-dependencies
 - HMM / PCFG models cannot model these
- Result: collateral damage to entire analysis
- Solution: Remove repairs prior to parsing
- Parse accuracy: 88% F oracle edits

Evaluating How MDE Affects Parsing



EVAL 1

Impact of Structural Metadata on Parsing

	Human Transcriptions	ASR Output
Human-Annotated Metadata	Best for the Parser	Use alignments
System-generated Metadata	<i>How much do metadata errors affect parsing accuracy?</i>	Worse for the Parser
Pause-duration Metadata	<i>How about poorer metadata models?</i>	Even Worse for the Parser

Overall Impact of Structural Metadata (SUs and EDITs) on Parsing (Charniak's parser on dev2)

	SU boundary	SU+subtype	Edit Words
Human:	27.30	36.89	53.39
ASR:	37.34	47.03	76.03

Bracketed F-measure	Human Transcriptions	ASR Output
Human Metadata	88.06	76.55
System Metadata	74.34	64.03

Progress on WSJ PTB

Model	Year	F-Measure
PTB	1993	
Magerman	1995	84.1
Collins	1996	85.5
Charniak	1997	86.6
Collins	1999	88.2
Charniak	2000	89.5
Collins	2000	89.7
Charniak & Johnson	2005	91.0

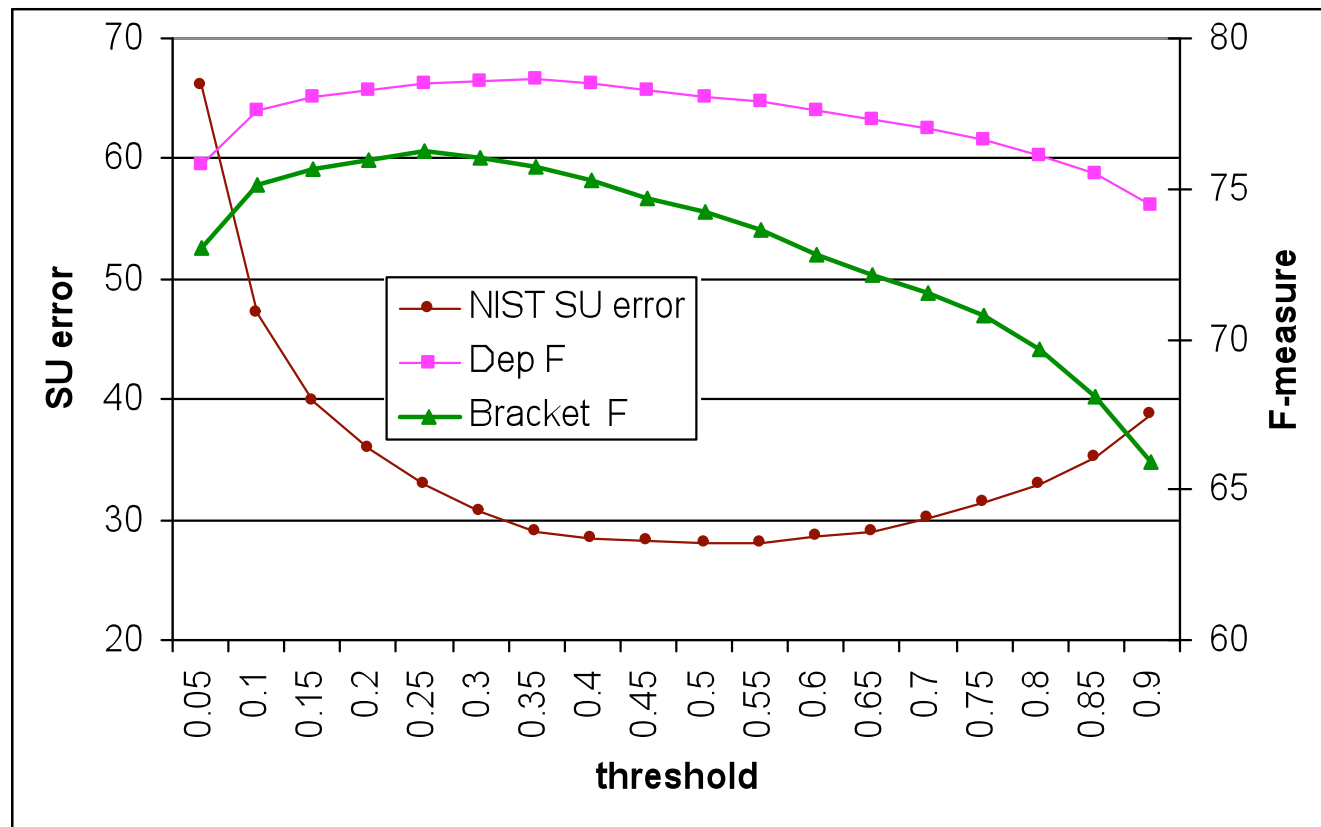
Impact of SUs and EDITs on Parsing (Charniak's parser on dev2) on Human Transcriptions

Bracketed F-measure	Human EDITs	System EDITs
Human SUs	88.06	83.25
System SUs	77.84	74.34

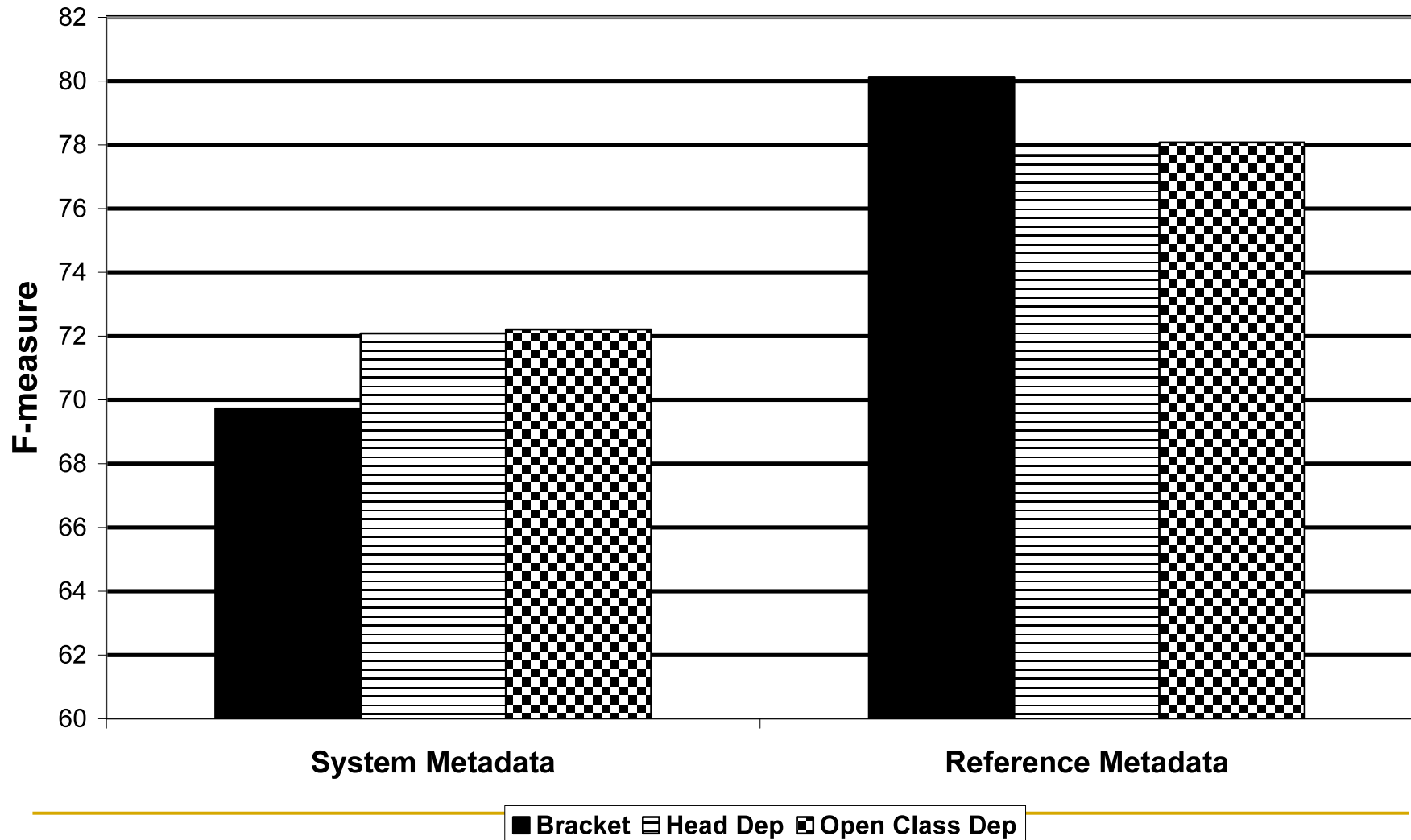
Impact of Different SU Detection Systems on Parsing (Charniak's parser on dev2)

Bracketed F-measure	Human Transcriptions	ASR Output
Human SUs	83.25	71.42
System SUs	74.34	64.03
Pause-based SUs (0.5s)	63.09	54.62

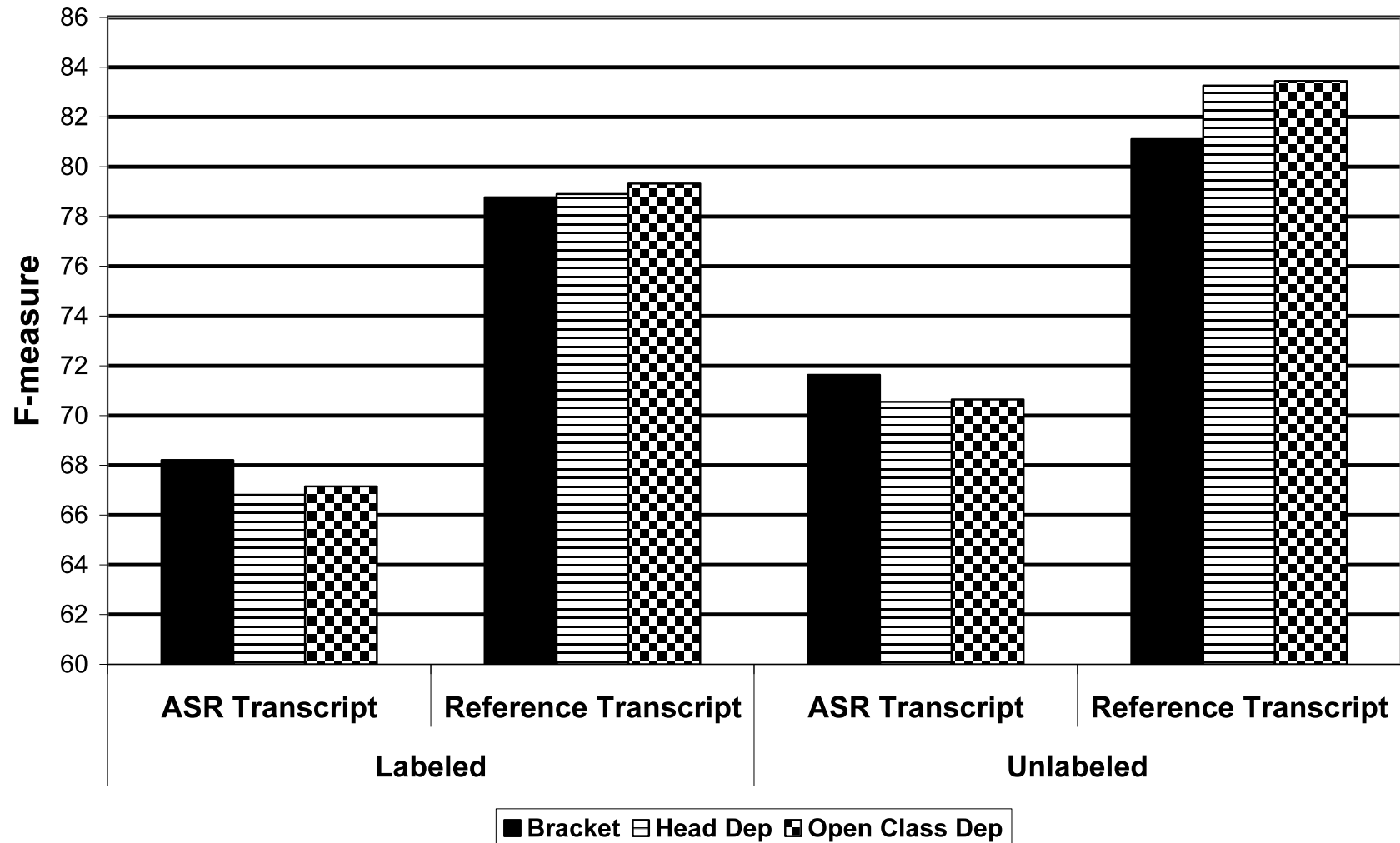
Impact of SU Threshold on Parsing Accuracy and SU Error



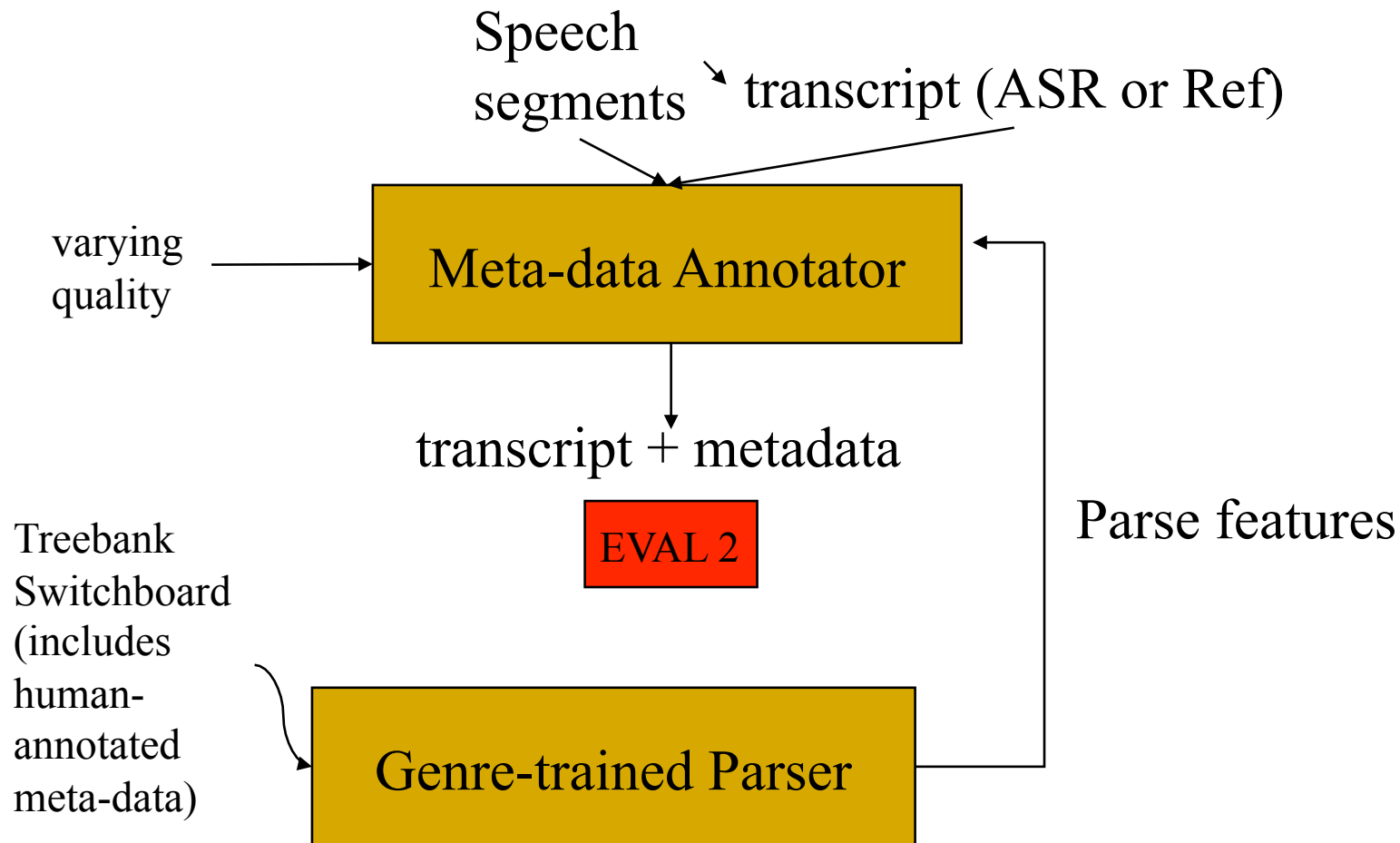
Metadata Quality X Parse Match Representation



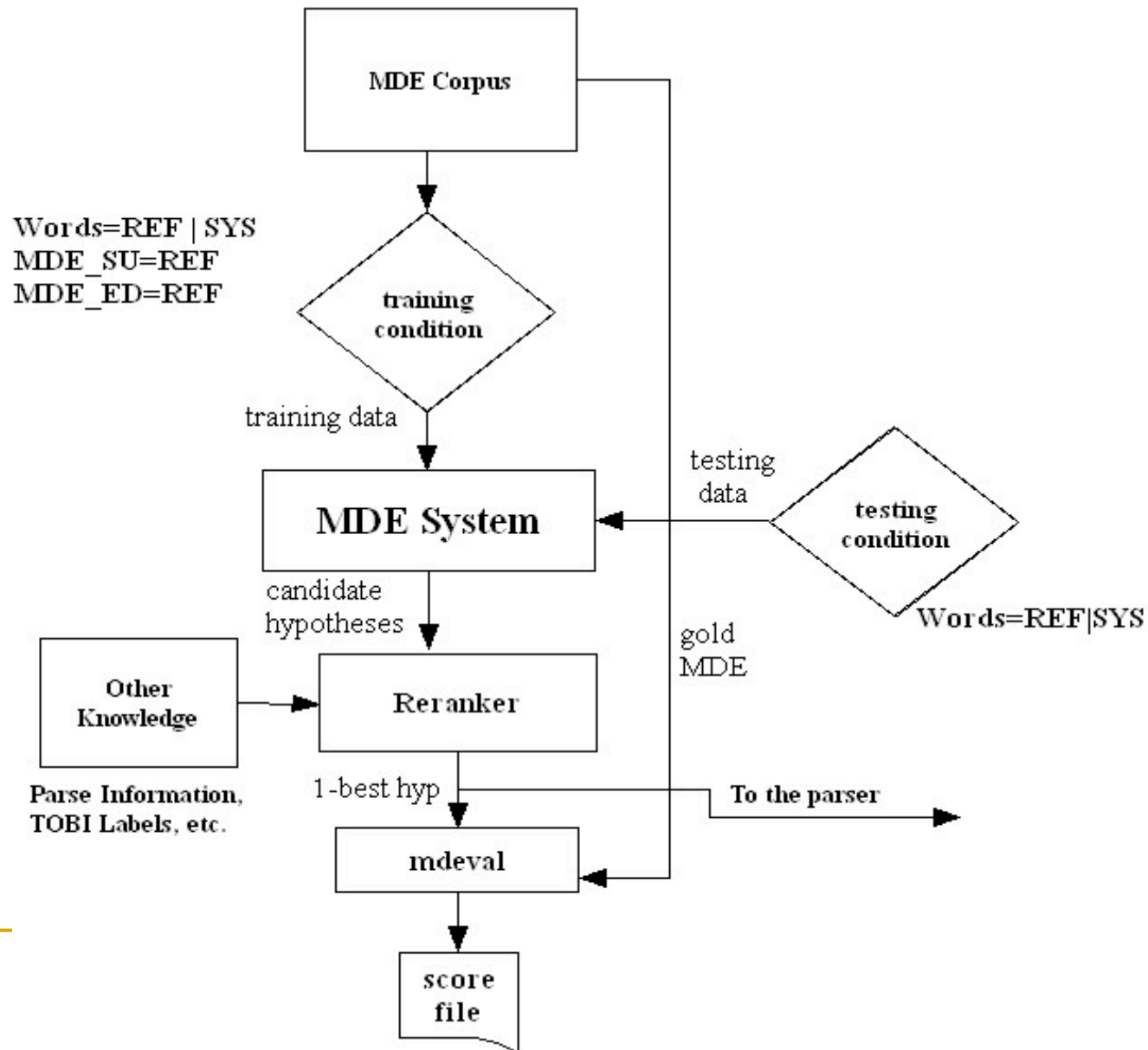
Labeling X Transcript Quality X Parse Match Representation



How does Parsing affect MDE?



Better Metadata Detection



Example features: tip of the iceberg

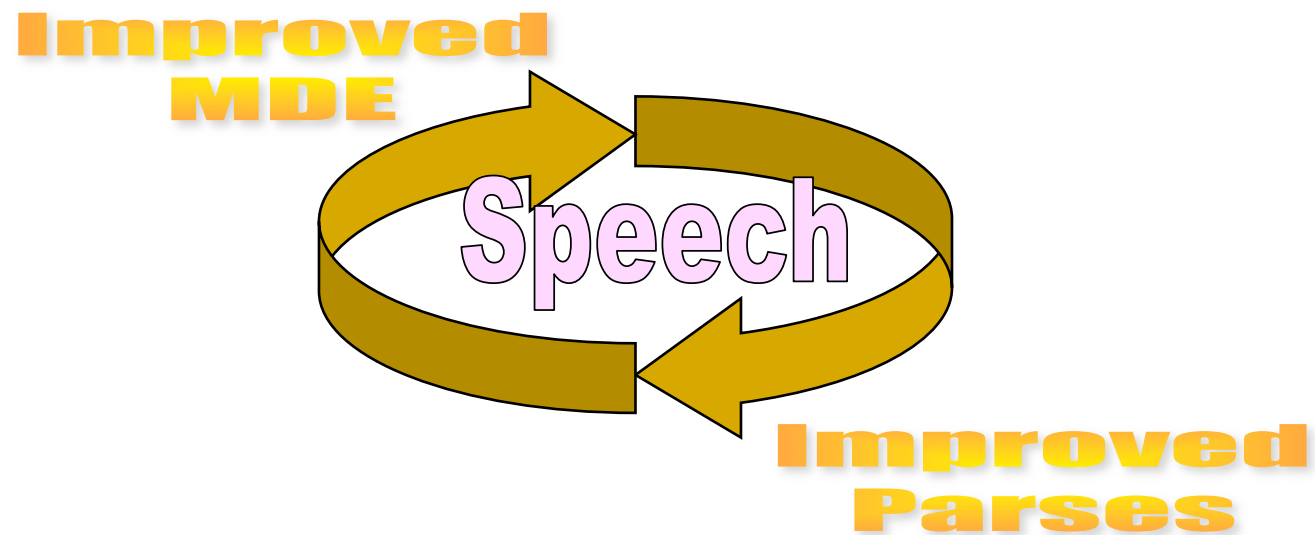
- posterior from baseline
- number of field internal segments guessed
- max/min segment lengths
- average segment length
- n-gram score
- Charniak parser LM score
- root symbol of Charniak viterbi trees
- root symbol + number of children of Charniak viterbi trees
- non-root symbols of viterbi trees
- non-root symbols + no. of children of viterbi trees
- Initial and final unigrams/bigrams
- Initial and final unitags/bitags
- Speaker change/backchannel indicators
- Baseline annotated disfluency information
- Constraint-Dependency Grammar (CDG) Parser-derived features
- Extracted dependency features from Charniak parser and Minipar
- TOBI based prosodic labels

Parse Accuracy with Reranking Experiments

System	Optimized for	SU performance				Bracketing F-measure	H-Dep F-measure
		P	R	F	NIST		
Baseline REF		87.2	82.7	84.9	29.4	74.0	77.3
Reranked REF	SU	86.9	86.7	86.8	26.4	76.3	78.7
Reranked REF	Parse	83.8	87.9	85.8	29.1	76.9	79.1

Baseline STT		83.3	77.7	80.4	37.9	63.9	65.8
Reranked STT	SU	84.2	78.7	81.3	36.1	64.8	66.4
Reranked STT	Parse	80.8	81.6	81.2	37.9	65.7	66.8

Explore the Synergy



Further Information

- Download Sparseval tool:

<http://www.clsp.jhu.edu/ws2005/groups/eventdetect/files/SParseval.tgz>

- See the workshop final report:

<http://www.clsp.jhu.edu/ws2005/groups/eventdetect/documents/finalreport.pdf>
