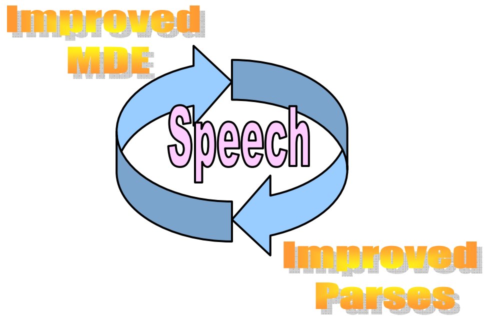
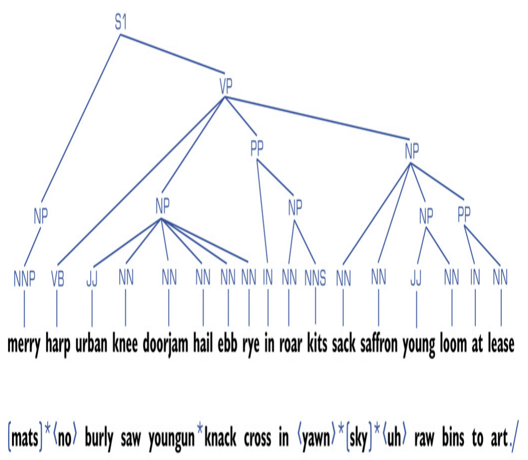
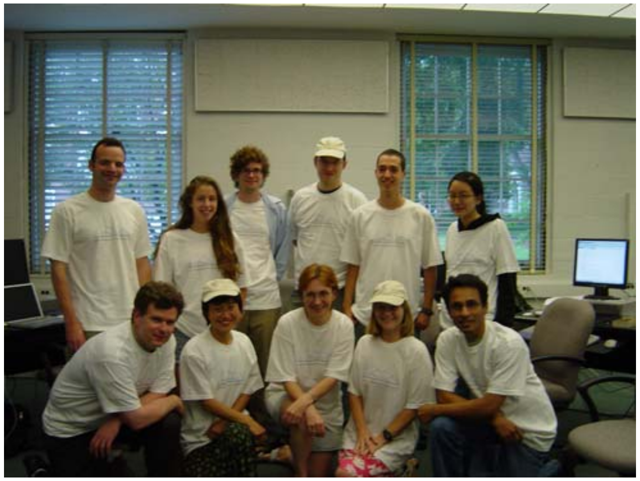


JHU CLSP 2005: Parsing and Structural Metadata in Speech

Where Parsing Meets Speech and Metadata Makes it Possible





Outline

- Background and Baseline Metadata Extraction
- Parsing Metrics and Impacting Factors
- Prosodic Structure
- Using Structural Knowledge to Improve Parsing
- Proposal: Disfluency and Parsing (Matt Lease)
- SU Reranking Experiments
- Proposal: Off-topic Detection (Robin Stewart)

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

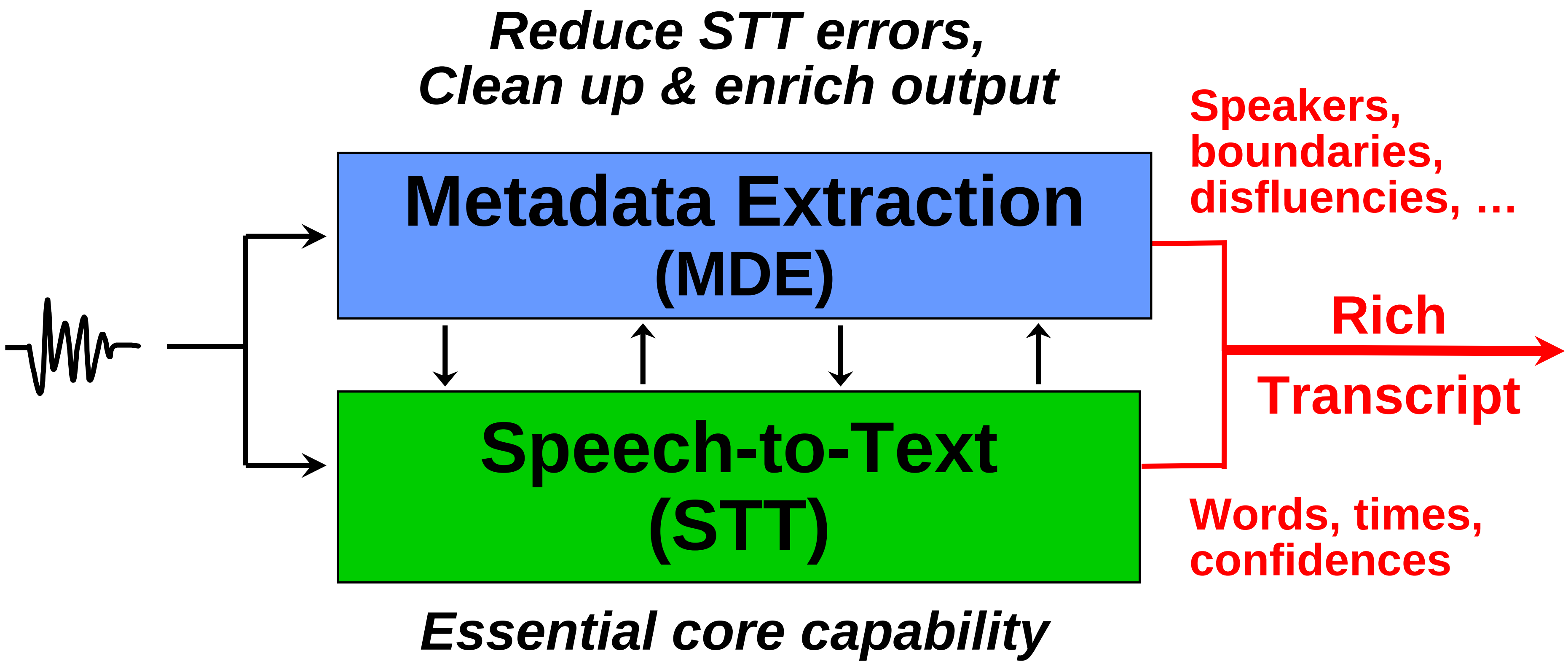
Issues in Language Processing Using Speech Recognition Output

- Segmentation issues:
  - Sentence boundaries are NOT provided and ASR segments are inappropriate
  - Parsing systems have a polynomial time complexity in the number of words
- Word strings contain:
  - ASR errors (insertions, deletions, and substitutions)
  - Phenomena atypical of textual sources (e.g., filled pauses, speech repairs)

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 /.

Synergistic Processes in EARS

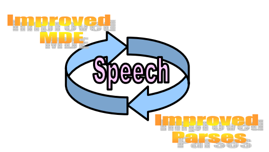


EARS Structural Metadata Extraction Tasks

- **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
- **Edit word detection:** reparandum region of a speech repair (e.g., **[we have]** * we set a string of charges)
- **Interruption point (IP) detection**

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

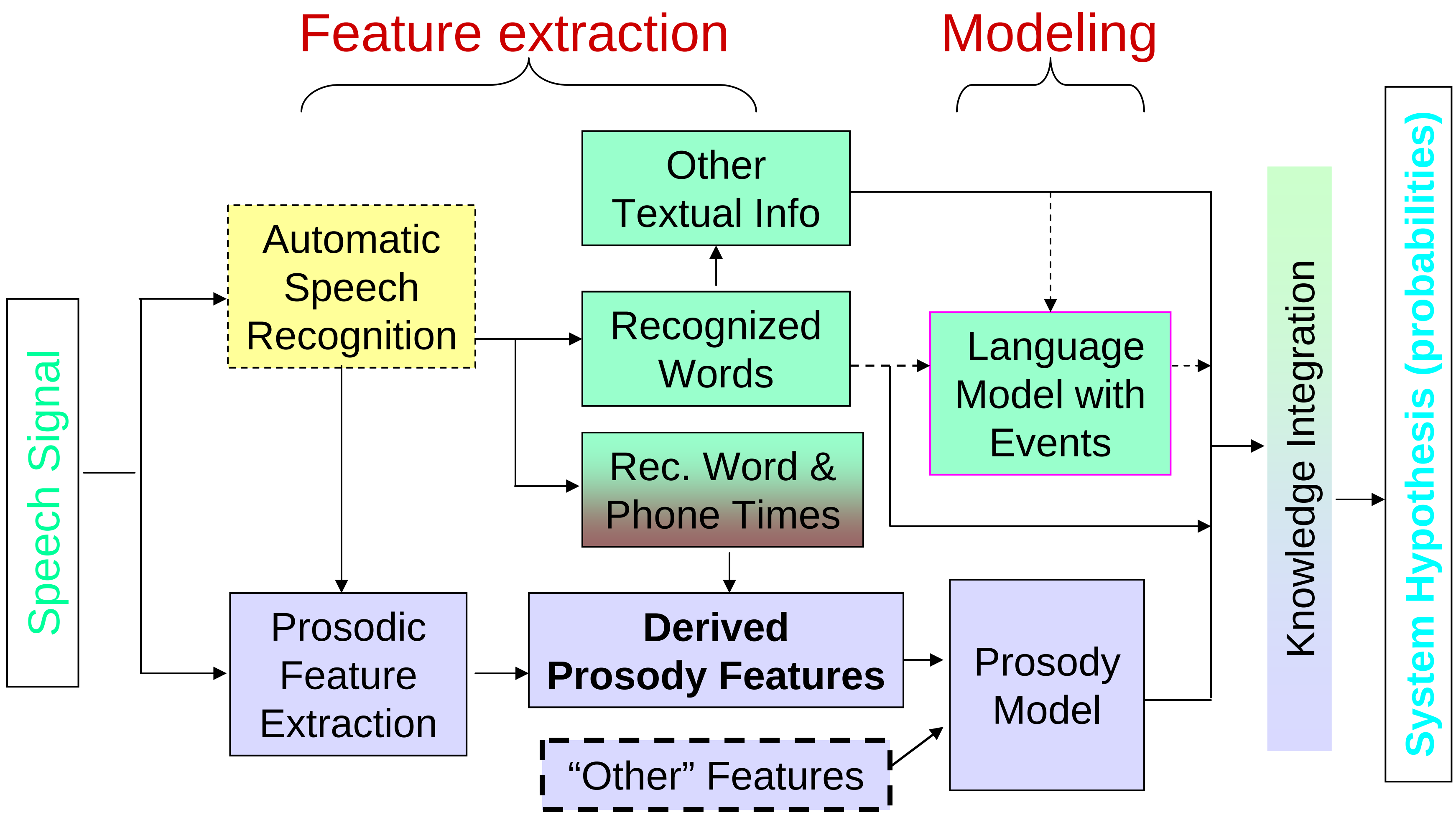


RT'04 Data Resources

- 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

General Modeling Framework in MDE (ICSI+SRI System)



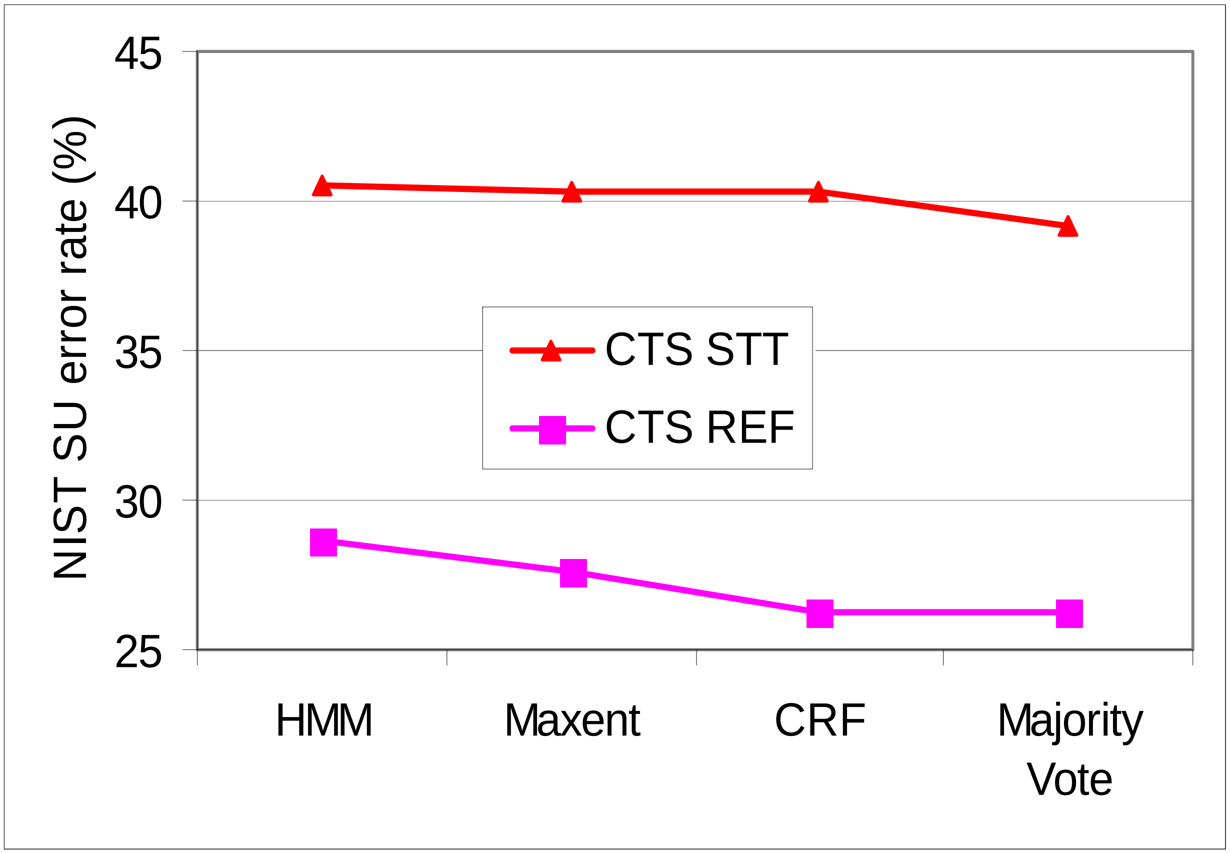
Summary of Modeling Approaches

	HMM	**Maximum Entropy (Maxent)**	**Conditional Random Fields (CRF)**
Discriminative training	N	Y	Y
Handles overlapping features	N	Y	Y
Models sequential information	Y	N	Y
Training is computationally efficient	Y	N	N

Features in Maxent and CRF

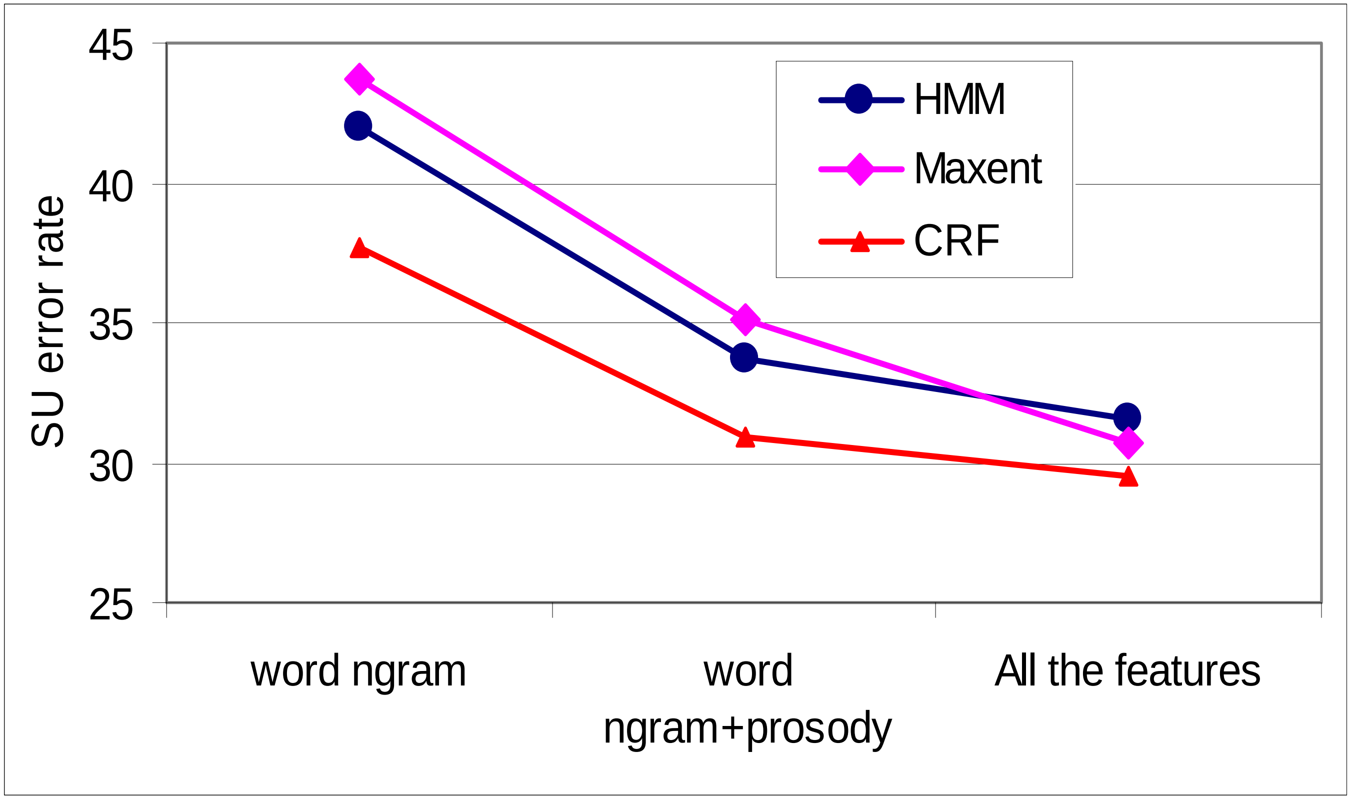
- Word N-grams
- Part-of-speech N-grams
- N-grams of automatically-induced class
- Cumulative binned posterior probabilities from the prosody model
- Cumulative binned posterior probabilities from the additional language models

SU Boundary Detection Results



NIST error rate = # errors / **# reference events**

SU Boundary Detection: Impact of Different Knowledge Sources



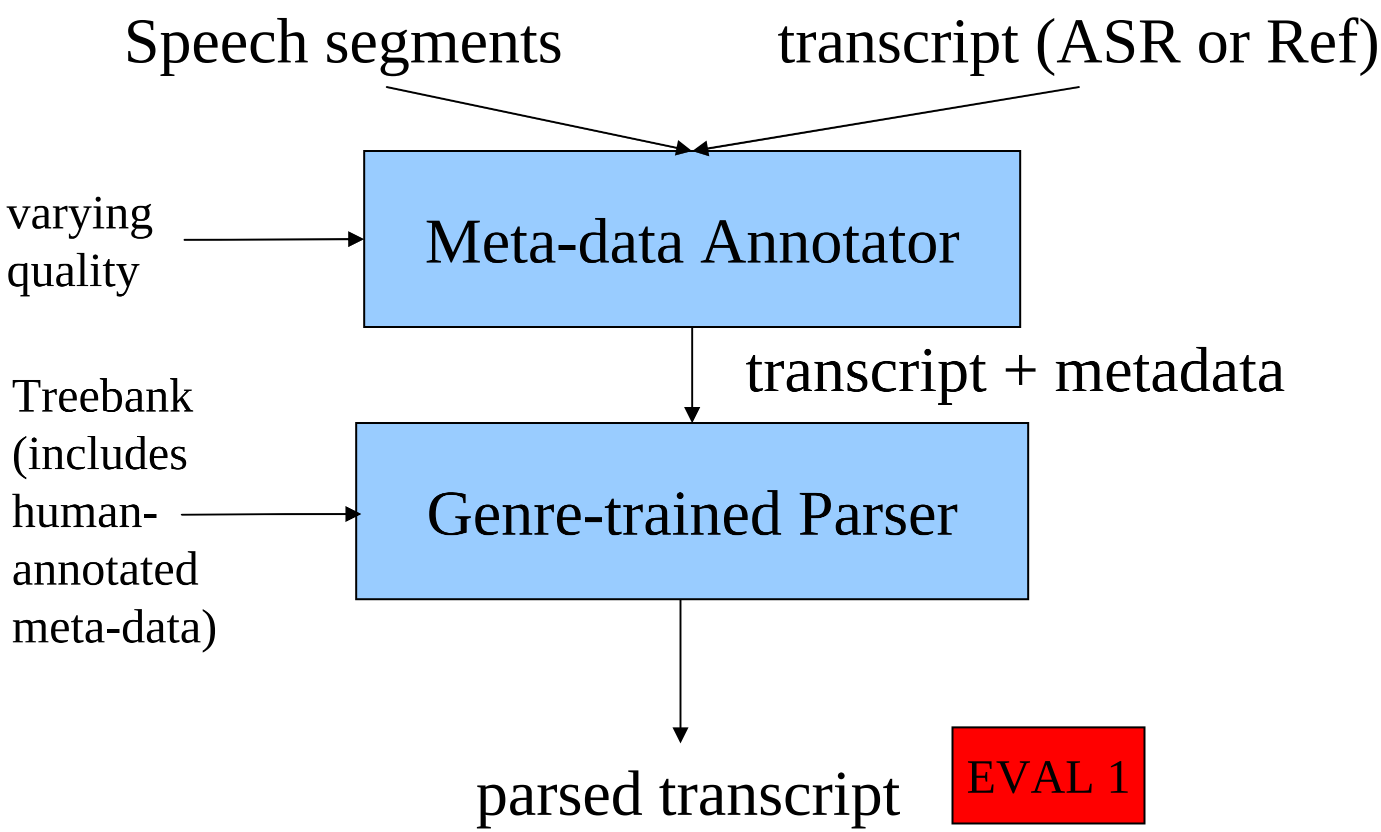
Remarks on Baseline MDE

- State-of-the-art metadata detection system
- Still much room for improvement !!!
  - Use reranking approach, good avenue to incorporate features
  - Folks at Brown University have used syntactic features for disfluency detection and achieved better results — motivation for using syntactic information in SU reranking
  - Note: in SU reranking, we use the posterior probabilities from the combination of HMM and Maxent systems
- We have made progress
  - Examined the impact of metadata on parsing
  - Incorporated many knowledge sources and improved metadata extraction

Roadmap

- Background and Baseline Metadata Extraction
- Parsing Metrics and Impacting Factors
- Prosodic Structure
- Using Structural Knowledge to Improve Parsing
- Proposal: Disfluency and Parsing (Matt Lease)
- SU Reranking Experiments
- Proposal: Off-topic Detection (Robin Stewart)

Evaluating How MDE Affects Parsing



Measuring Parse Accuracy on Speech

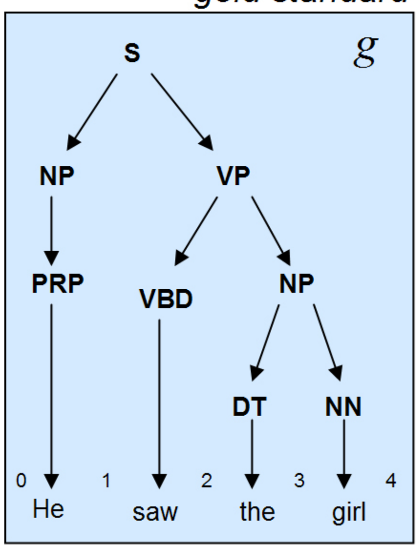
- How do we measure parsing accuracy given:
  - Word mismatch
  - SU mismatch
- Alignment:
  - Reference transcript and ASR output can be aligned
- Metrics investigated:
  - bracket-based (i.e., adapt Parseval metrics)
  - dependency-based

Parsing Metrics: Brackets

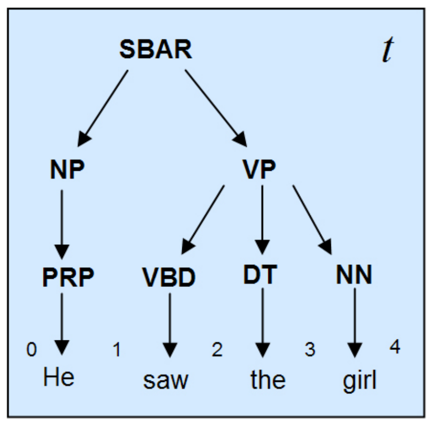
$$brackets(g) = \{S(0,4),\ \underline{NP(0,1)},\ \underline{VP(1,4)},\ NP(2,4)\}$$

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

gold standard



evaluation tree



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

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

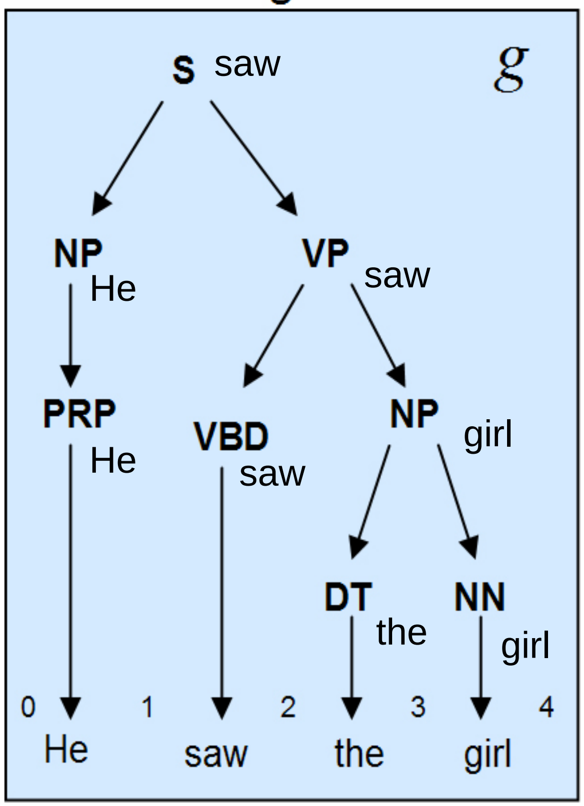
$$F_{meas}(t,\ g) = \frac{2 \cdot 66.66 \cdot 50}{66.66 + 50} = 57.14\%$$

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

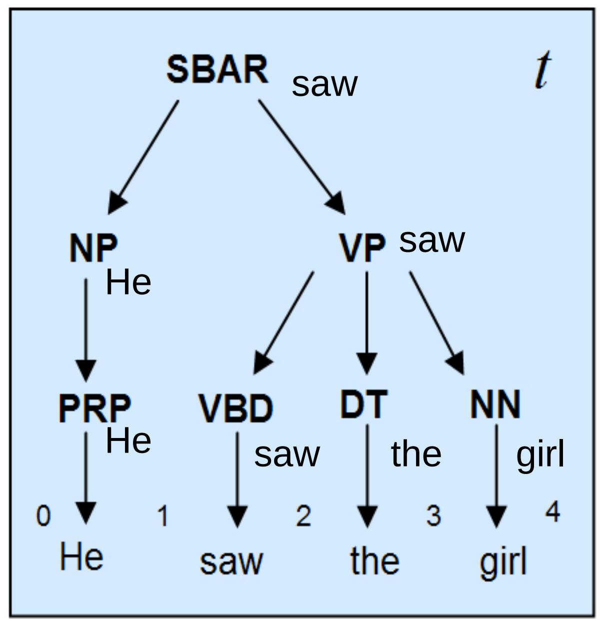
Parsing Metrics: Head Dependency

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)}

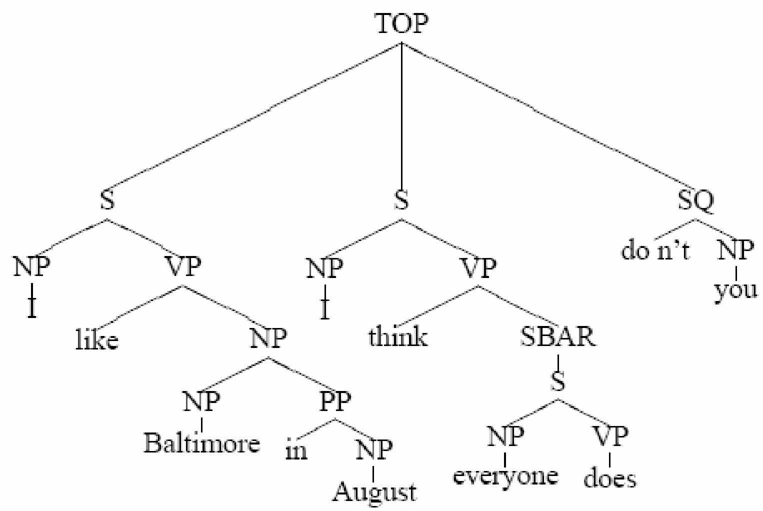
gold standard



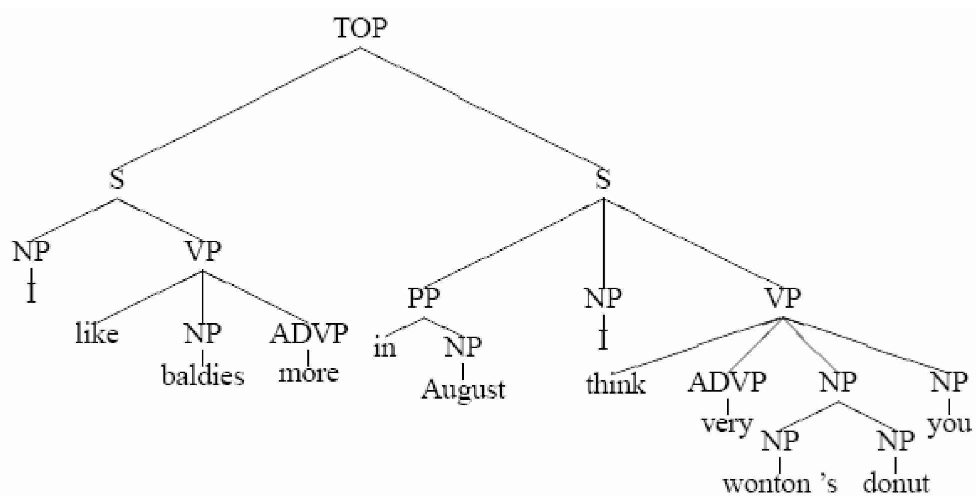
evaluation tree



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)

Matching Test to Gold Given Different Words and SUs on Conversation Side

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
Baltmore	baldies	001
	more	010
in	in	000
August	August	000
I	I	000
think	think	000
everyone	very	001
does	wonton	001
	’s	010
do	donut	001
n’t		100
you	you	000

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

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

The Parsing Metrics Evaluated

- Types:
  - Dependencies (words matter)
    - all dependencies versus open class only
    - head percolation rules (Charniak, Collins, Hwa)
    - use alignment or not
  - Brackets (alignment required)
- Other Conditions:
  - Labeled versus Unlabeled

Correlations in the Aligned Case Across Conditions and Parsers

X-Y Correlations	Recall	Precision	F-measure
Brackets – All Deps	0.89	0.87	0.88
All Deps – Open Deps	0.99	0.99	0.99

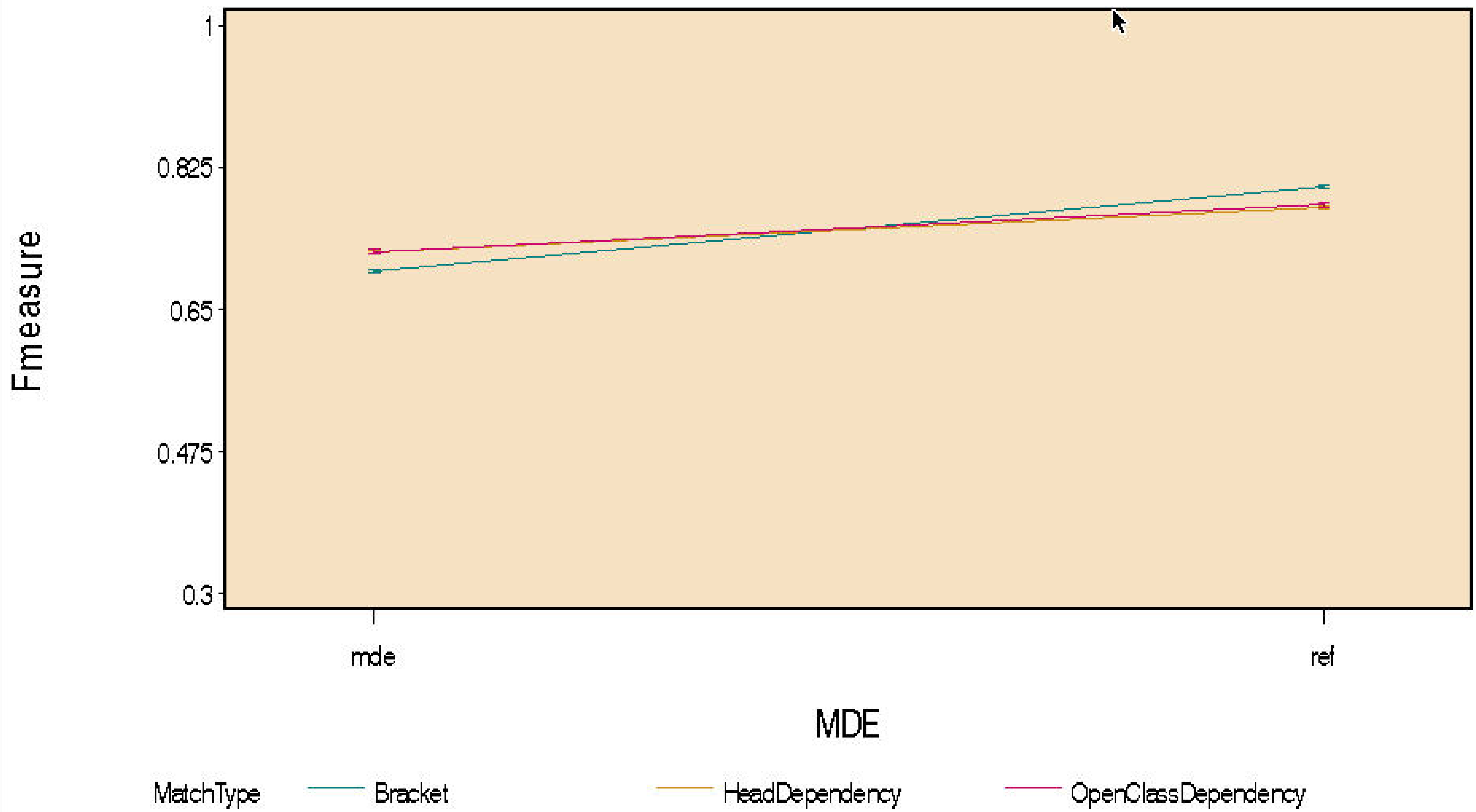
Statistical Analysis of Factors

- **Data Factors:**
  - **Transcription type:** Reference (ref) vs. STT (stt)
- **Algorithm Control Factors:**
  - **Parser:** Charniak, Bikel, Roark
  - **Metadata type:** Reference (ref) versus System (mde)
  - **EDIT MDE:** Use it or not
- **Parse Match Factors:**
  - **Match Type:** Bracket, Head Dependency, Open Class Dependency
  - **Conversation Side Word Alignment:** Used versus Not
  - **Labels:** Used versus Ignored
- **Dependent Measure:** F-measure (Precision and Recall are similar)

Significant Metric Main Effects

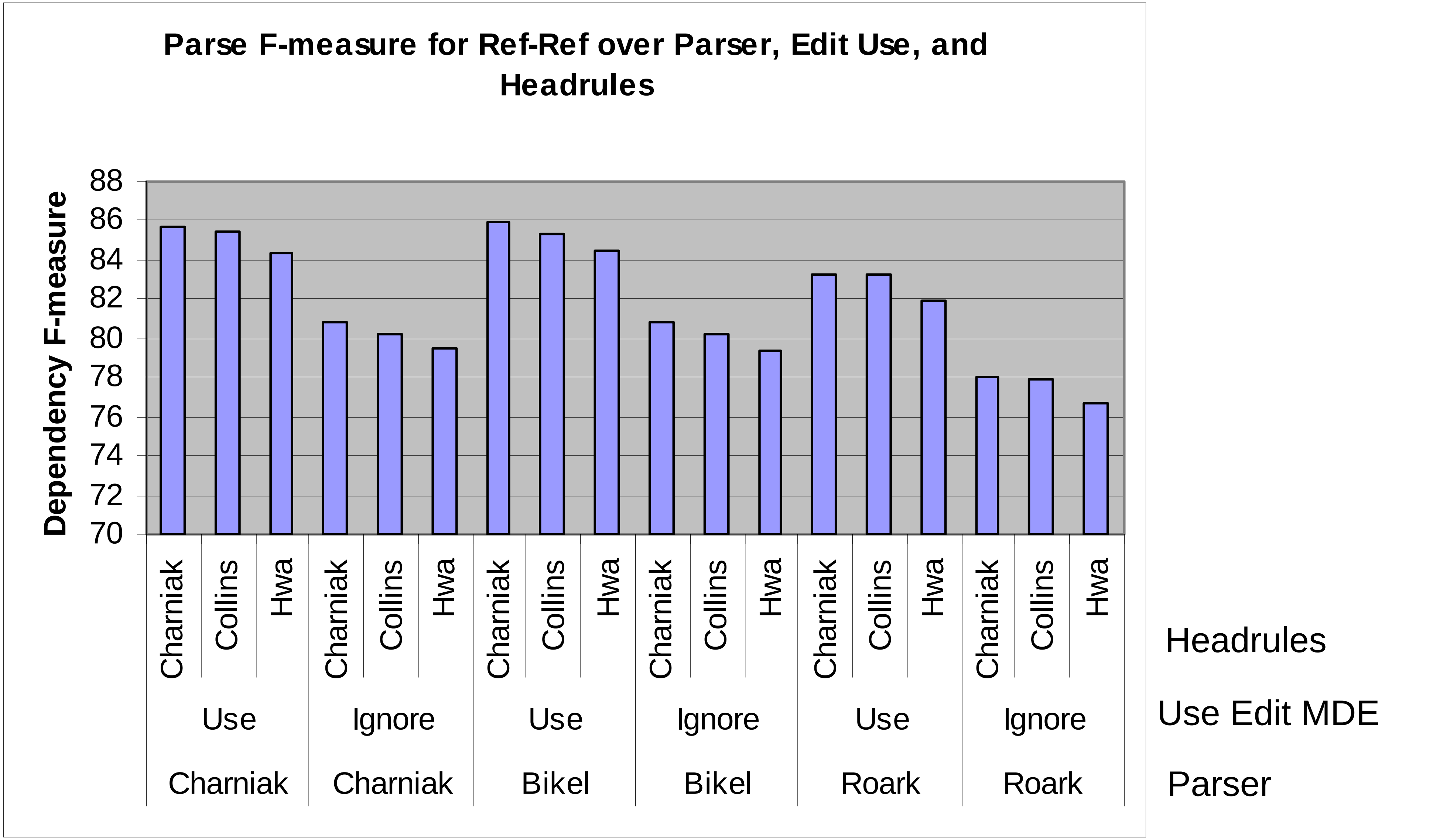
- **Labeling:** Unlabeled scores are significantly greater than labeled scores
- **Head Percolation Rules:** they matter when extracting dependencies to score all parsers (Charniak > Collins > Hwa)
- **MatchType:** All Dependencies, Open Class, Brackets
- **Alignment not significant**
- Some interesting significant interactions between match type and other factors (e.g., transcription type, labeling, MDE type)

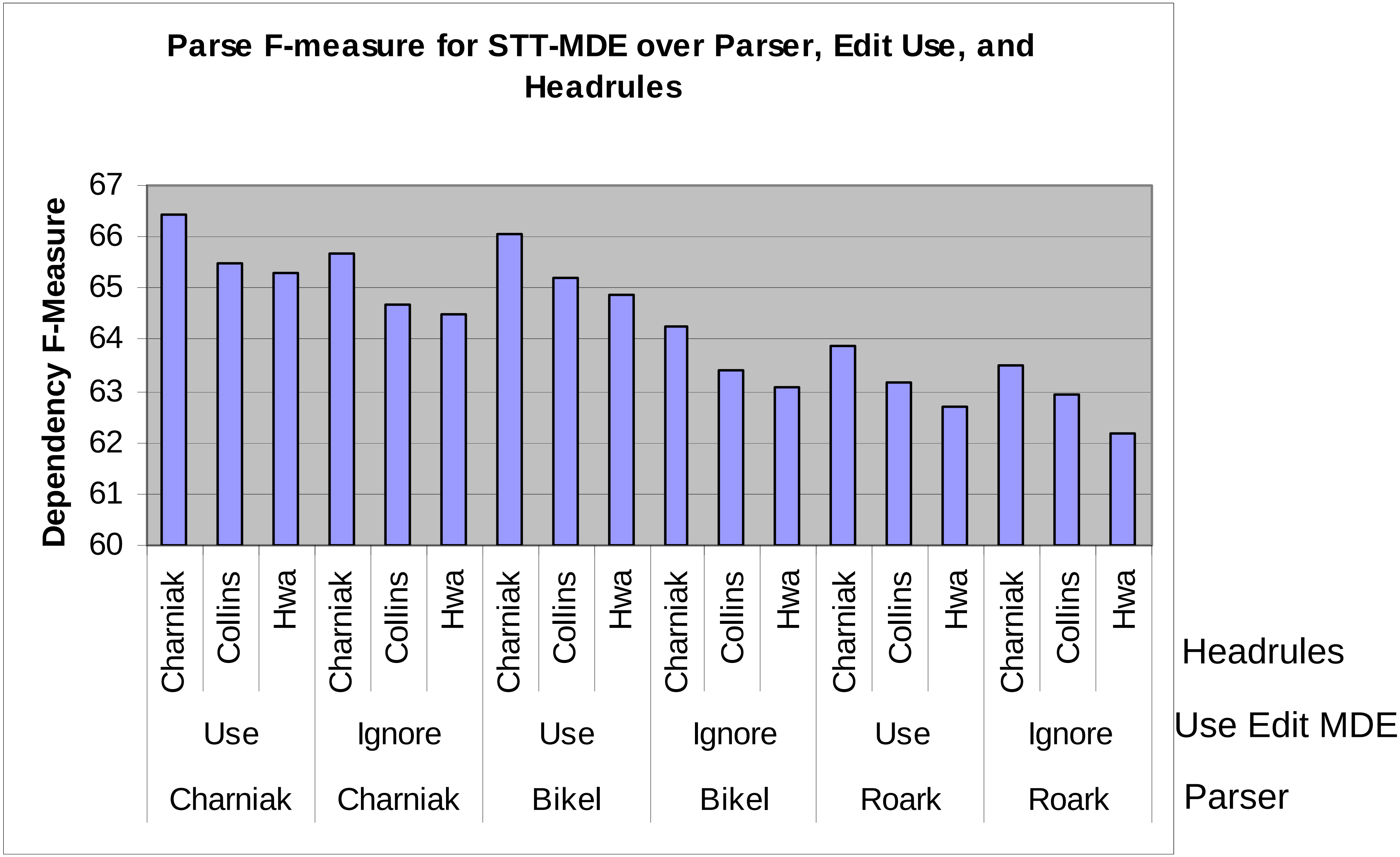
The Effect of MDE on Parsing Across Metrics



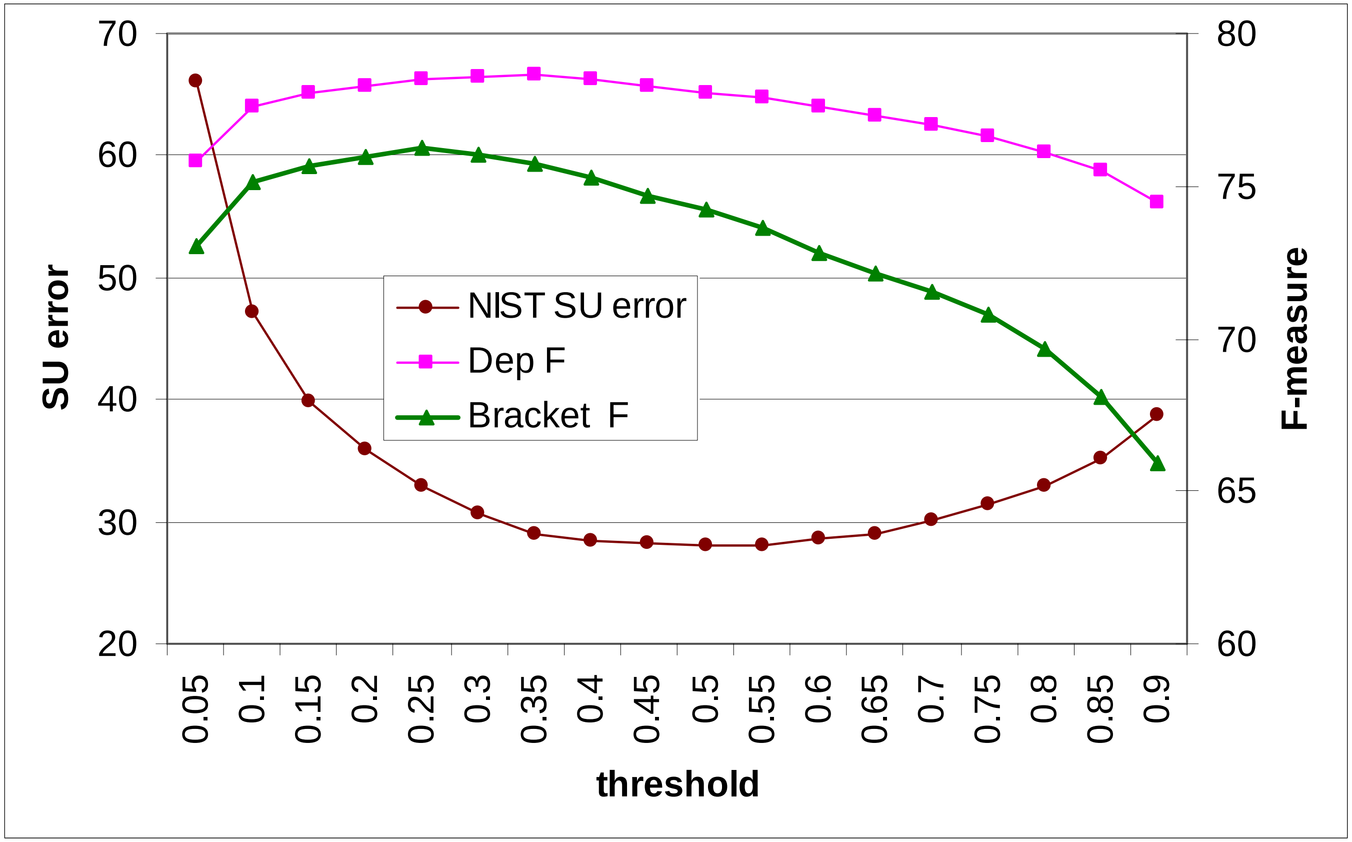
Significant Data Effects

- **Transcription:** ref > stt
- **MDE:** ref > mde
- **EDITs:** remove to parse > letting parser handle them
- **EDIT USE x MDE:** Using ref edits helps more than using mde edits
- **Parser x EDIT USE x MDE**

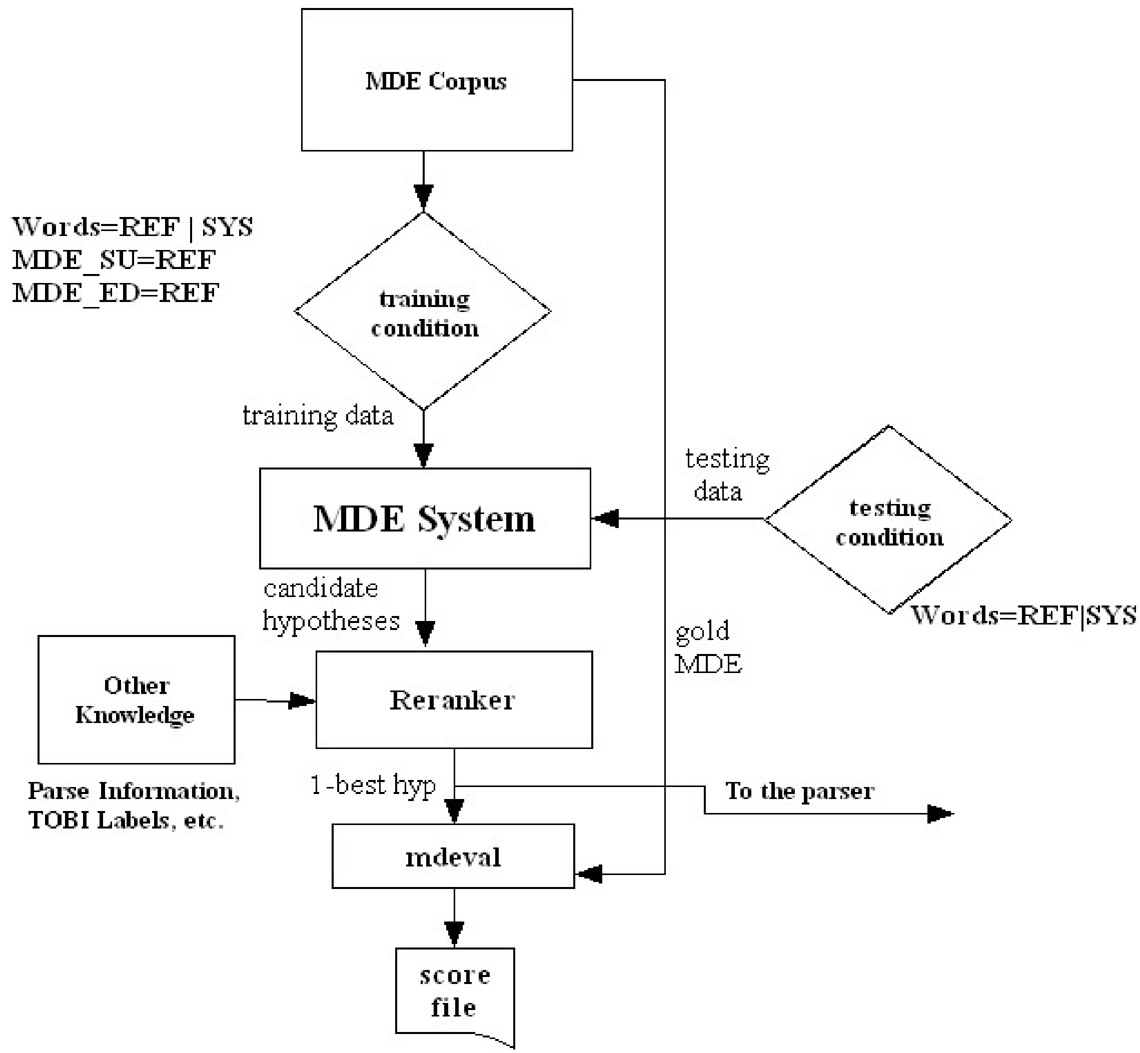

Parse F-measure for Ref-Ref over Parser, Edit Use, and Headrules
Dependency F-measure
88
86
84
82
80
78
76
74
72
70
Charniak
Collins
Hwa
Headrules
Use
Ignore
Use Edit MDE
Charniak
Bikel
Roark
Parser


Parse F-measure for STT-MDE over Parser, Edit Use, and Headrules
Dependency F-Measure
67
66
65
64
63
62
61
60
Charniak
Collins
Hwa
Charniak
Collins
Hwa
Charniak
Collins
Hwa
Charniak
Collins
Hwa
Charniak
Collins
Hwa
Charniak
Collins
Hwa
Use
Ignore
Use
Ignore
Use
Ignore
Charniak
Charniak
Bikel
Bikel
Roark
Roark
Headrules
Use Edit MDE
Parser

Impact of SU Threshold on Parsing Accuracy and SU Error



The Impact of Knowledge Sources on Metadata Detection



Roadmap

- Background and Baseline Metadata Extraction
- Parsing Metrics and Impacting Factors
- Prosodic Structure
- Using Structural Knowledge to Improve Parsing
- Proposal: Disfluency and Parsing (Matt Lease)
- SU Reranking Experiments
- Proposal: Off-topic Detection (Robin Stewart)

Prosodic Structure

Consider the utterance:

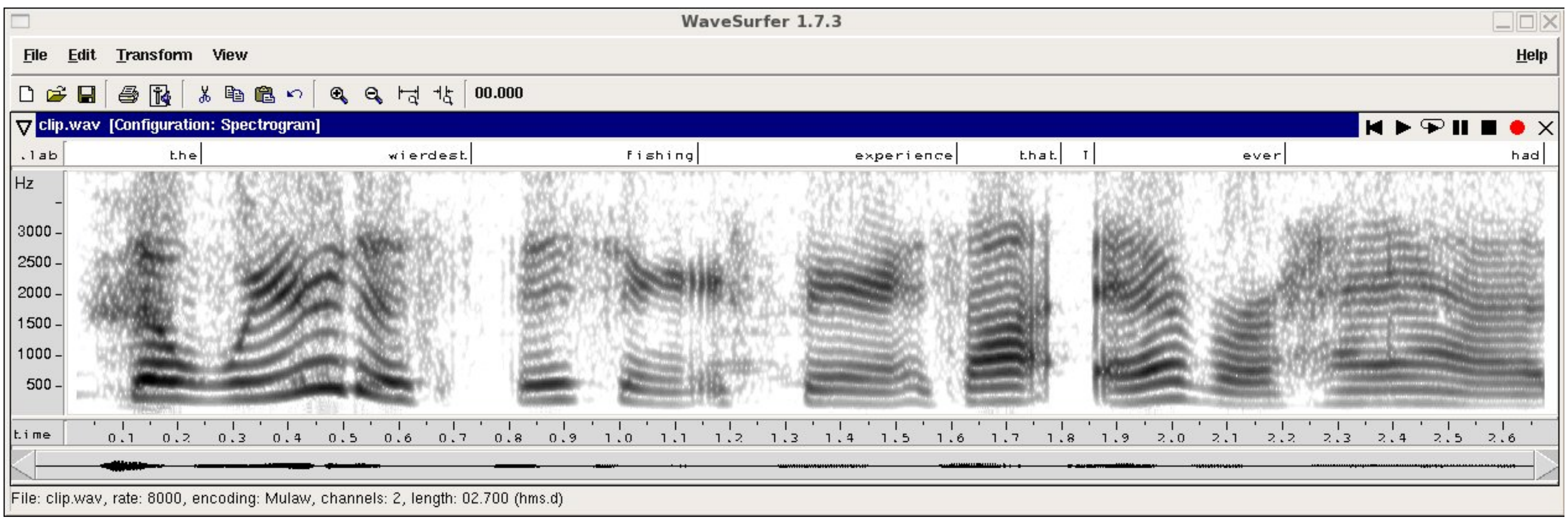
Spoken words: *the weirdest fishing experience i ever had people to this day are still trying to figure out if i really caught what i think i caught*

Prosodic Structure

Consider the utterance:

Spoken words: *the weirdest fishing experience i ever had people to this day are still trying to figure out if i really caught what i think i caught*

But, there is more info in speech: **(a) pitch excursions in** *weirdest*, **(b) loudness variations, and (c) syllable lengthening in** *had*.



Prosodic Structure

- **Tones: Create contrasts via pitch variations, and highlight associated words or phrases.**
- **Breaks: Segment speech into groups of syllables or words.**

Prosodic Structure

- **Tones**: Create contrasts via pitch variations, and highlight associated words or phrases.
- **Breaks**: Segment speech into groups of syllables or words.
- **Tones and Break Indices, ToBI** (Silverman et al, 1992). Utterance $\approx$ a sequence of minor or intermediate phrases, embedded in major or intonational phrases ($\sim$ clauses).

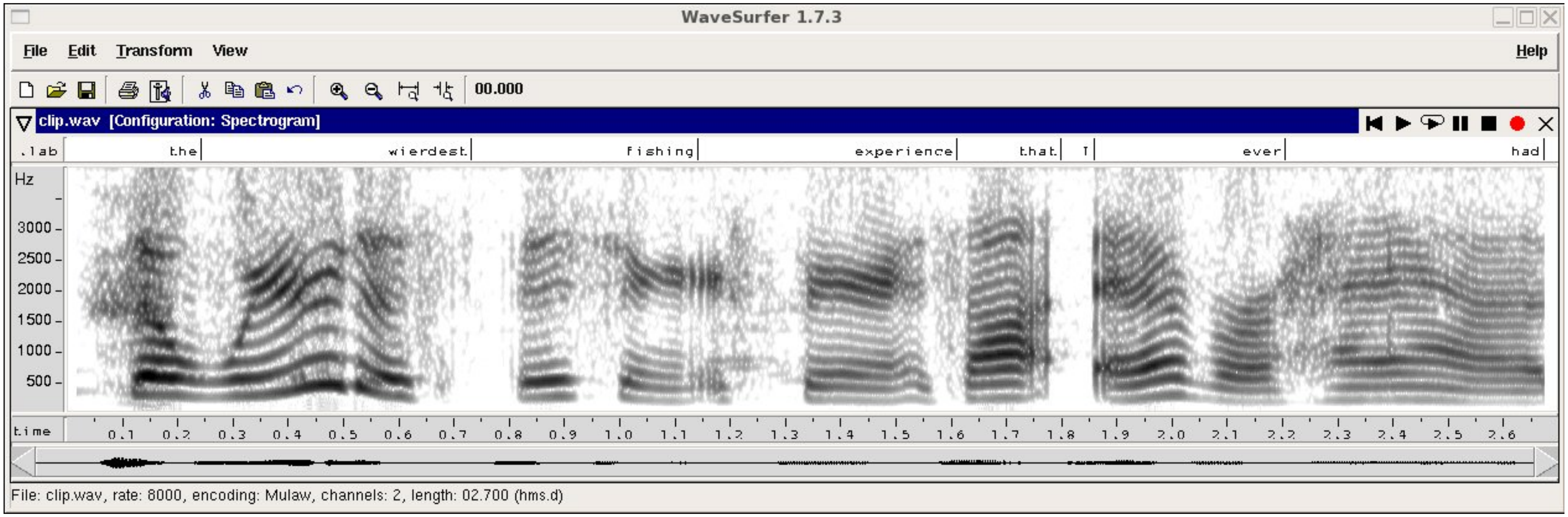
Prosodic Structure

- Tones: Create contrasts via pitch variations, and highlight associated words or phrases.
- Breaks: Segment speech into groups of syllables or words.
- Tones and Break Indices, ToBI (Silverman et al, 1992). Utterance $\approx$ a sequence of minor or intermediate phrases, embedded in major or intonational phrases ($\sim$ clauses).
- Note, there are alternative schemes without embedding, e.g., Utterance $\approx$ sequence of prosodic phrases called *f-groups* obtained using, what they call, *chinks 'n chunks algorithm* (Liberman and Church, 1992).

Prosodic Structure

- **Tones: Create contrasts via pitch variations, and highlight associated words or phrases.**
- **Breaks: Segment speech into groups of syllables or words.**
- **Tones and Break Indices, ToBI (Silverman et al, 1992). Utterance $\approx$ a sequence of minor or intermediate phrases, embedded in major or intonational phrases ($\sim$ clauses).**
- **Note, there are alternative schemes without embedding, e.g., Utterance $\approx$ sequence of prosodic phrases called** *f-groups* **obtained using, what they call,** *chinks 'n chunks algorithm* **(Liberman and Church, 1992).**
- **Fortunately, we have a small conversational speech corpus with ToBI labels – a 64 conversation subset of SWB (Ostendorf et al 2000).**

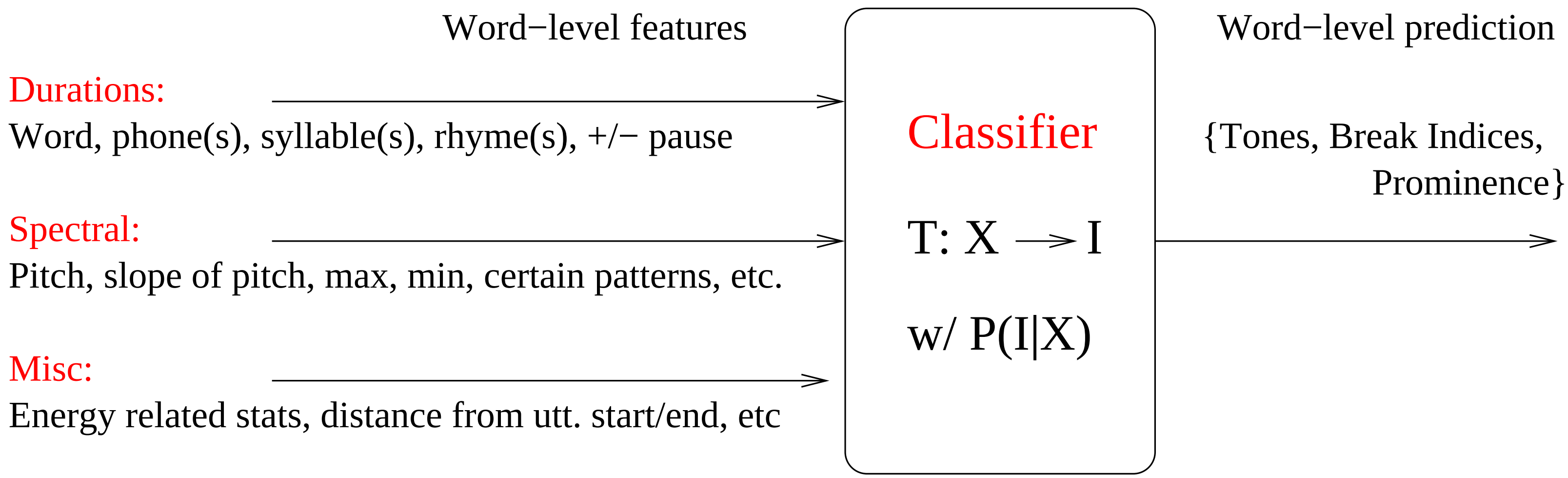
ToBI Annotation Scheme



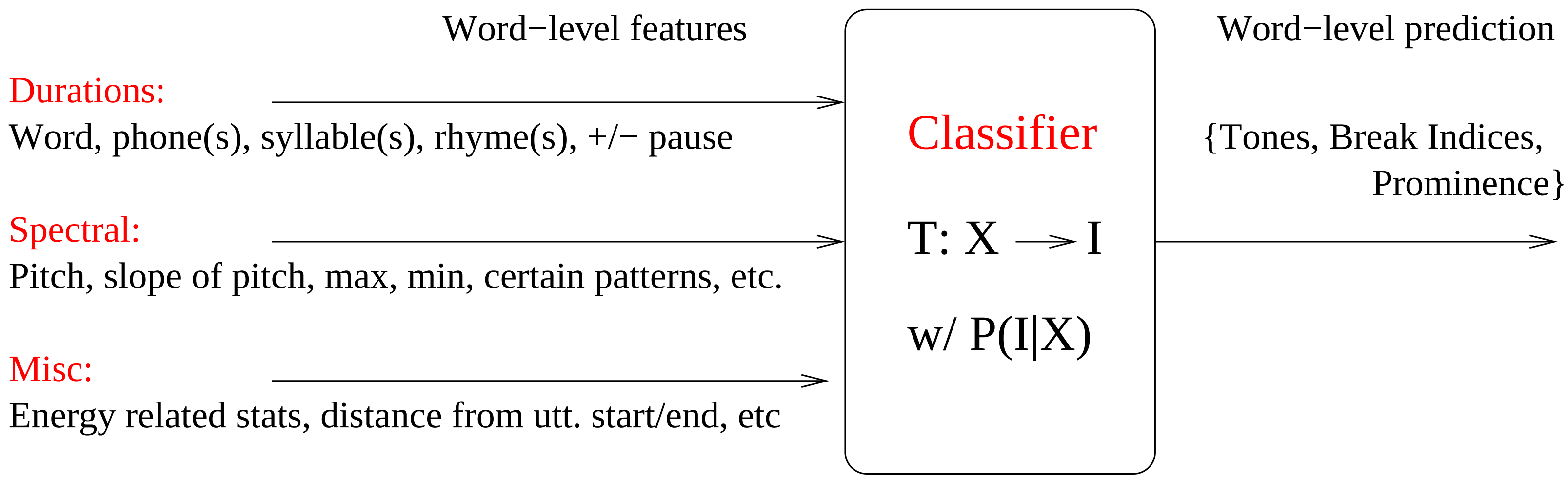
the /1/ weirdest /1,/ fishing /1/ experience /3,L-/ i /1/ ever /3,*,L-/ had /4,*.L-H%/ uh /2p/*

1. **Break Indices: 0, 1, 2, 3, 4, 1-, 2-, 3-, 4-.** **[collapsed to 1,4]**
2. **Disfluency: 1p, 2p, 3p.** **[collapsed to p]**
3. **Tones: H-H%, H-L%, L-L%, L-H%, H-, L-.**
4. **Prominence: ***

Classifier


Word−level features
Durations:
Word, phone(s), syllable(s), rhyme(s), +/− pause
Spectral:
Pitch, slope of pitch, max, min, certain patterns, etc.
Misc:
Energy related stats, distance from utt. start/end, etc
Classifier
T: X → I
w/ P(I|X)
Word−level prediction
{Tones, Break Indices,
Prominence}

Classifier



- **Feature $\sim$ Y. Liu et al's baseline MDE system. (Shriberg et al 2000, Sonmez et al 1999)**
- **Features don't use word or phone identity, hence likely to be useful when transcript are unreliable, as in ASR.**
- **Decision tree-based classifier using IND, apt to deal w/ missing features (e.g. pitch).**

Classification Results

a) Breaks: 81.7% (67.7%)

	1	4	p
1	33665	922	524
4	3492	6032	1217
p	1693	1673	2679

b) Prominence: 78.9% (67.5%)

	absent	present
absent	37664	5482
present	8039	12754

Classification Results

a) Breaks: 81.7% (67.7%)

	1	4	p
1	33665	922	524
4	3492	6032	1217
p	1693	1673	2679

b) Prominence: 78.9% (67.5%)

	absent	present
absent	37664	5482
present	8039	12754

- Good performance on break indices and prominence.
- Accuracy of tones is low at 53.3% (41.1%).

Classification Results

a) Breaks: 81.7% (67.7%)

	1	4	p
1	33665	922	524
4	3492	6032	1217
p	1693	1673	2679

b) Prominence: 78.9% (67.5%)

	absent	present
absent	37664	5482
present	8039	12754

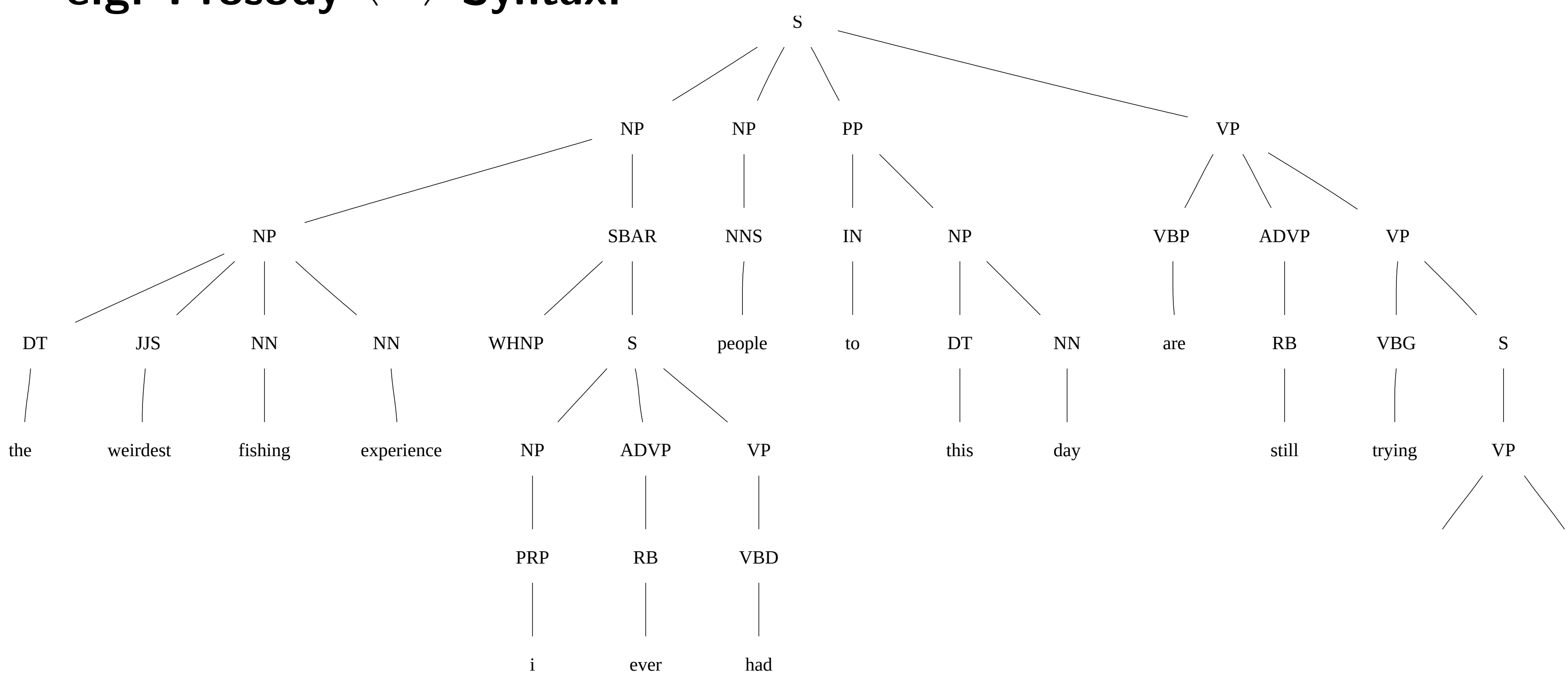
- Good performance on break indices and prominence.
- Accuracy of tones is low at 53.3% (41.1%).
- Simple extensions on break indices.
  - Temporal Markov constraints does not have much impact.
  - Voting improved performance (82.4%) marginally.
- Baseline classifiers (a) and (b) were used from hereon.

Task Overview

Team Goal: Explore the synergy between syntax and metadata.
e.g. Prosody $\Longleftrightarrow$ Syntax.

Task Overview

Team Goal: Explore the synergy between syntax and metadata.
e.g. Prosody $\Longleftrightarrow$ Syntax.



Prosodic Breaks: *the weirdest fishing experience /3/ i ever had /4/ people to this day /4/ are still trying to figure out*

Metadata Tasks

Metadata tasks can be seen as projections of the parsing problem.

1. SU detection: find the boundaries of the root constituent.
2. EDIT detection: find the boundaries of an EDITED constituent.
3. FILLER detection: find the boundaries of a FILLER constituent.

Can Prosodic Structure Help in SU Detection?

Expectation: Prosody groups syllables, alternatively, segments speech. Absence of prosodic break implies fluent region, and this reduces the possibility of SU boundary ($\sim$ Cutler et al 1997).

Can Prosodic Structure Help in SU Detection?

Expectation: Prosody groups syllables, alternatively, segments speech. Absence of prosodic break implies fluent region, and this reduces the possibility of SU boundary ($\sim$ Cutler et al 1997).

Simple experiment: Augment baseline SU detection system w/ posterior probability of ToBI labels.

NIST SU Error on Fisher-dev2:

	Baseline	w/ Breaks	w/ Proms	w/ Brks+Proms
Ref words	27.36	27.32	27.01	26.71
ASR words	35.78	35.52	35.30	34.90

Note, we see gain in ASR condition, even though the raw prosodic cues are already present in the baseline system.

Can Prosodic Structure Help in SU Detection?

- Independent of word or phone identity, can potentially generalize better when transcript is less reliable.
- Complex segment level features can be computed. Stay tuned to see how this benefits the re-ranking expts.
- Taking this further, can prosodic structure help parsing and associated metadata task, e.g. edit detection.

Metadata in Parsing Spoken Language

The possible space of exploring the synergy between syntax and metadata includes the following.

1. Enrich input to parsing.
2. Enrich the grammar itself.
3. Detect the words in edited region, excise them, parse the rest and then recombine (e.g. Johnson & Charniak 2004).

But, (3) relates speech repairs to syntactic structure only indirectly.

Prosody and Syntax

- The prosody-syntax interface is an active area of (psycho)linguistic research (Selkirk 1984, Nespor and Vogel 1986, Steedman 2000, Butt 1998).
- We tried a simple and direct interface using (1) and (2).

.

Prosody and Syntax

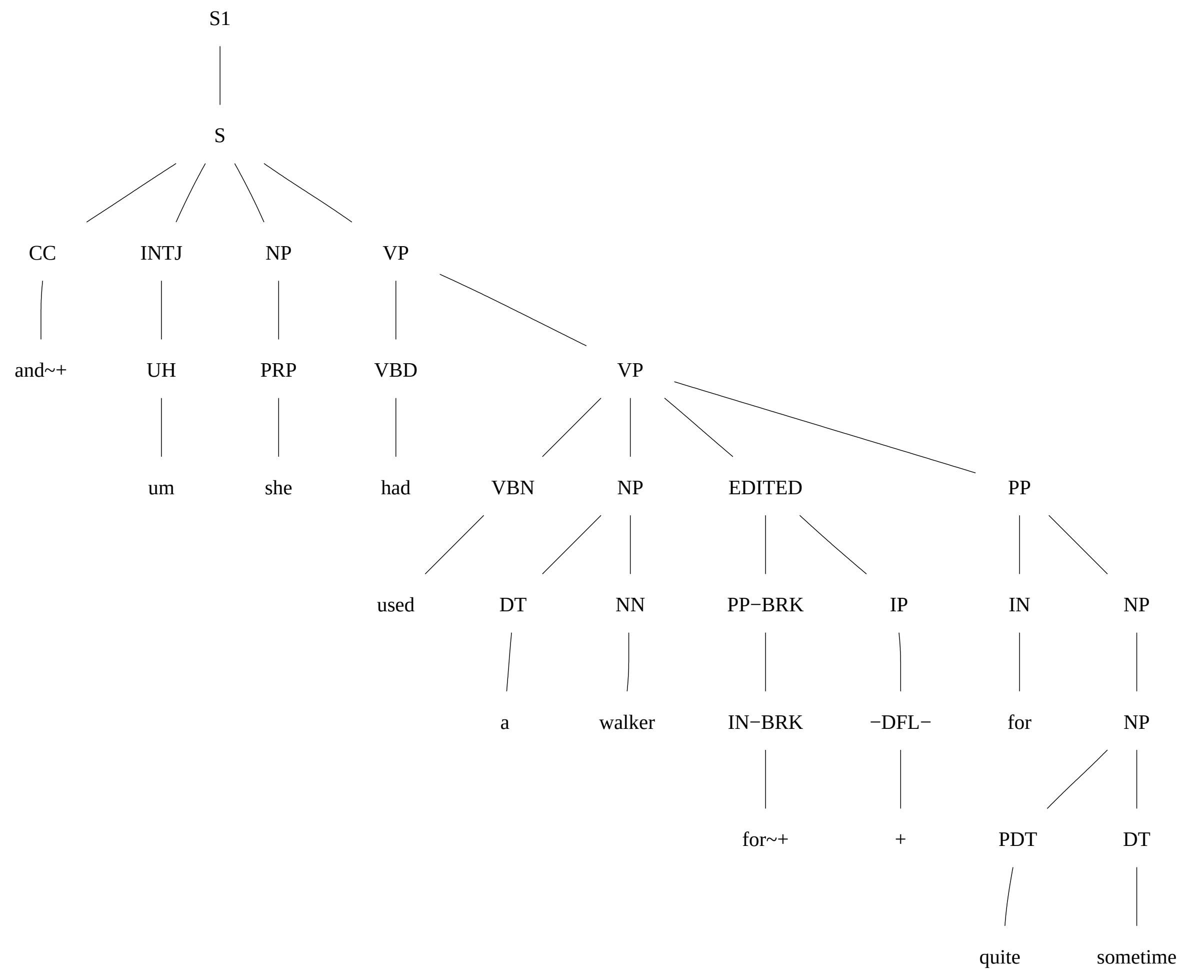
- The prosody-syntax interface is an active area of (psycho)linguistic research (Selkirk 1984, Nespor and Vogel 1986, Steedman 2000, Butt 1998).
- We tried a simple and direct interface using (1) and (2).

.

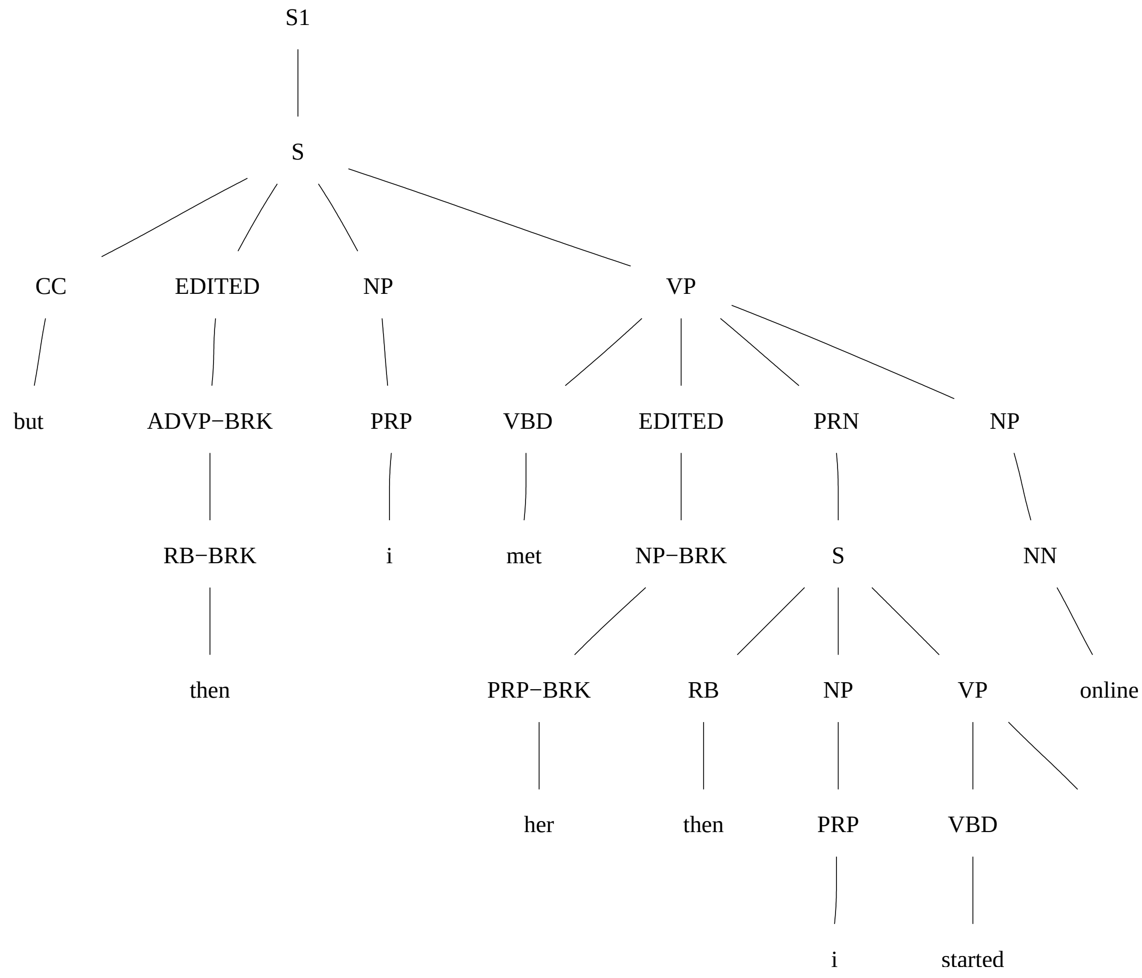
Hypothesis: diacritic 'p' cues edit (akin to Lickley 1996).

- Train: SWB with gold POS tags and automatic 'p'.
- The errors in prosodic tag are modeled as noise in a PCFG.
- Test: Fisher Dev2 with automatic POS tags and 'p' using CKY.

Correct Correlation



Overgeneralization



Evaluation

	PARSECAL	Per Word F-measure for Edits
Baseline	67.66	21.5
w/ breaks	64.89	30.6

- Edit detection improves, however, hurts overall performance.
- Rem: 'p' is more abundant than its syntactic counterpart – 78% recall, but only 30% precision.
- There are other disfluency $\Longleftrightarrow$ correlations that are profitable, which will be described shortly.

Roadmap

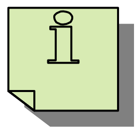
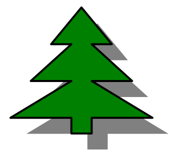
- Background and Baseline Metadata Extraction
- Parsing Metrics and Impacting Factors
- Prosodic Structure
- Using Structural Knowledge to Improve Parsing
- <u>Proposal:</u> Disfluency and Parsing (Matt Lease)
- SU Reranking Experiments
- <u>Proposal:</u> Off-topic Detection (Robin Stewart)

Two Mismatch Fixers

- **Enrich input** with a description of the change needed to make a more fluent version

- **Enrich grammar** to cover disfluent constructions as well

Improving Parsing for Speech

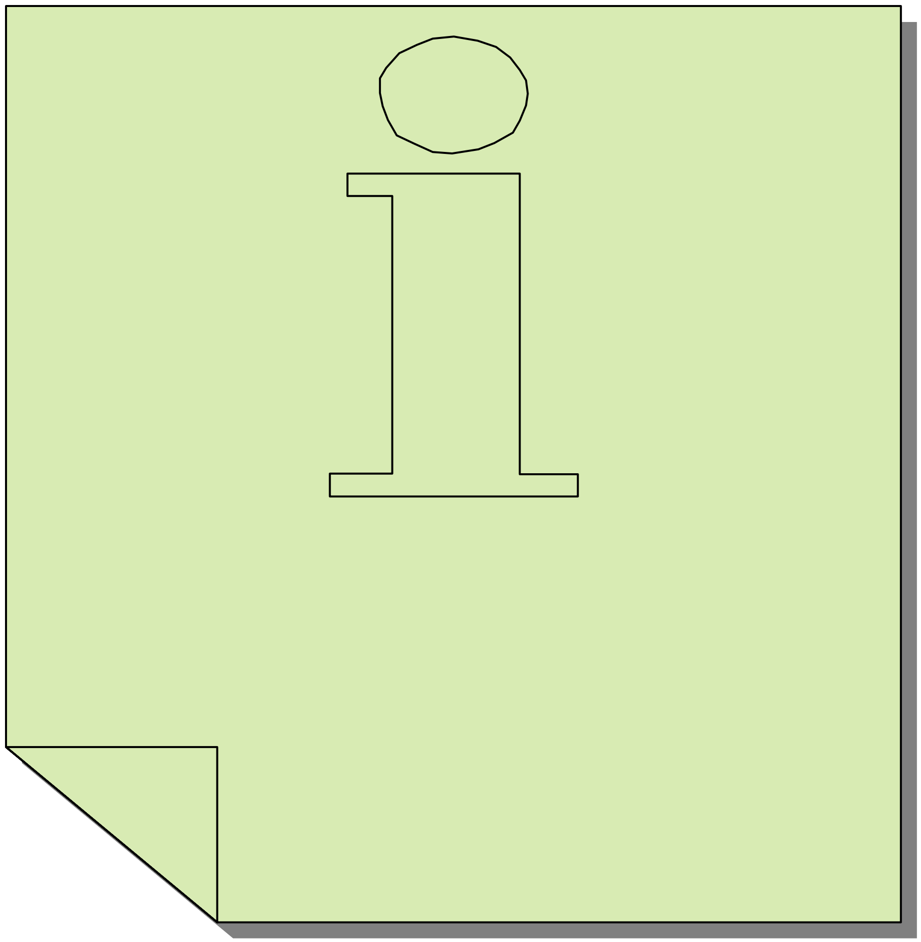
How do  ,  → Parsing ?

- Trained vs. Untrained Parsing
- Headed vs. Bracketed Evaluation

Roadmap

- Using a Minimalist parser to interpret marked up input string
  - REF: Humans provided annotations
  - MDE: Annotations automatically assigned (Liu, 2005)
- Modify conventional PCFG for disfluency
  - Unfinished phrases
  - Syntactic parallelism in speech repairs
- Evaluation
  - Test impact of markup on parser
  - Use bag of heads to overcome sentence-boundary error

Enrich Input



Automatic EDIT / FILLER markup

MDE Annotations automatically assigned using prosodic and lexical features (Liu, 2005)

INPUT: … `and I uh you know I guess as a young kid …`

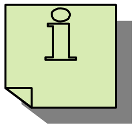
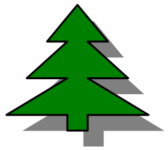
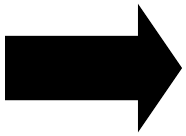
ENRICHED INPUT: …and <EDIT_ST> I <EDIT_END> <FL_ST> uh <FL_END> <FL_ST> you know <FL_END> <EDIT_ST> I <EDIT_END> I guess as a young kid…

Two Mismatch Fixers

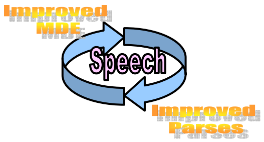
- **Enrich input** with a description of the change needed to make a more fluent version

- **Enrich grammar** to cover disfluent constructions as well

Improving Parsing for Speech

How do  ,   Parsing ?

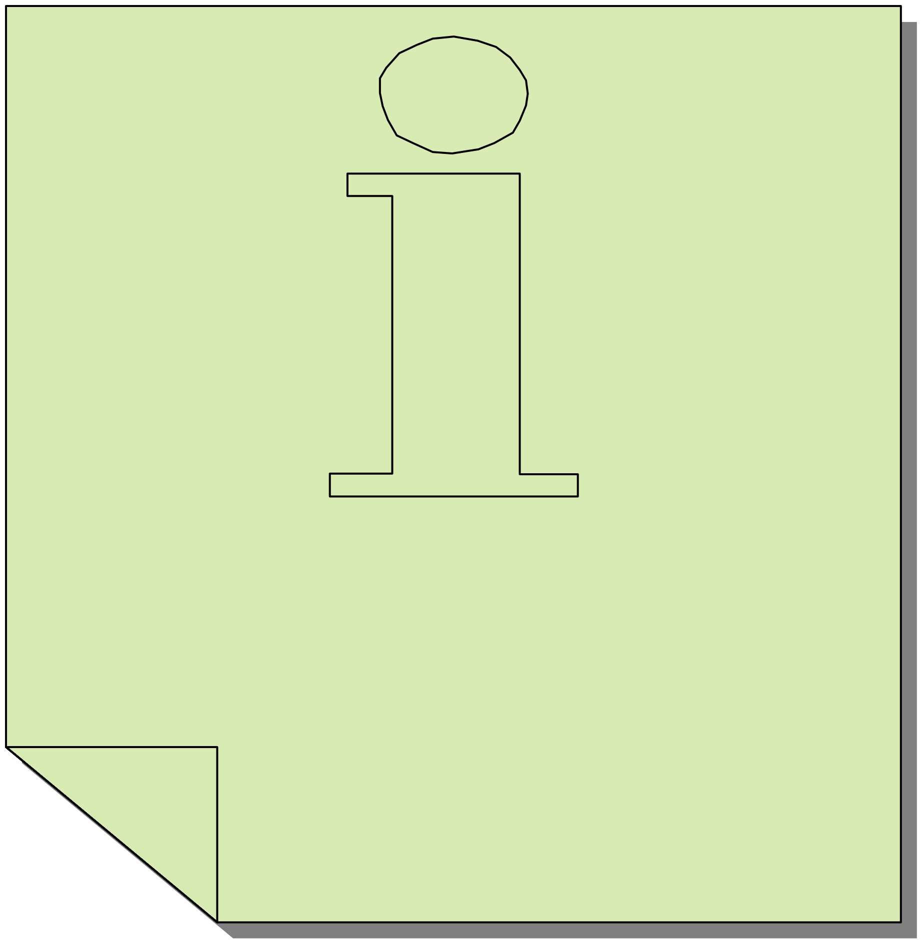
- Trained vs. Untrained Parsing
- Headed vs. Bracketed Evaluation



Roadmap

- Using a Minimalist parser to interpret marked up input string
  - REF: Humans provided annotations
  - MDE: Annotations automatically assigned (Liu, 2005)
- Modify conventional PCFG for disfluency
  - Unfinished phrases
  - Syntactic parallelism in speech repairs
- Evaluation
  - Test impact of markup on parser
  - Use bag of heads to overcome sentence-boundary error

Enrich Input



Automatic EDIT / FILLER markup

MDE Annotations automatically assigned using prosodic and lexical features (Liu, 2005)

INPUT: … **`and I uh you know I guess as a young kid …`**

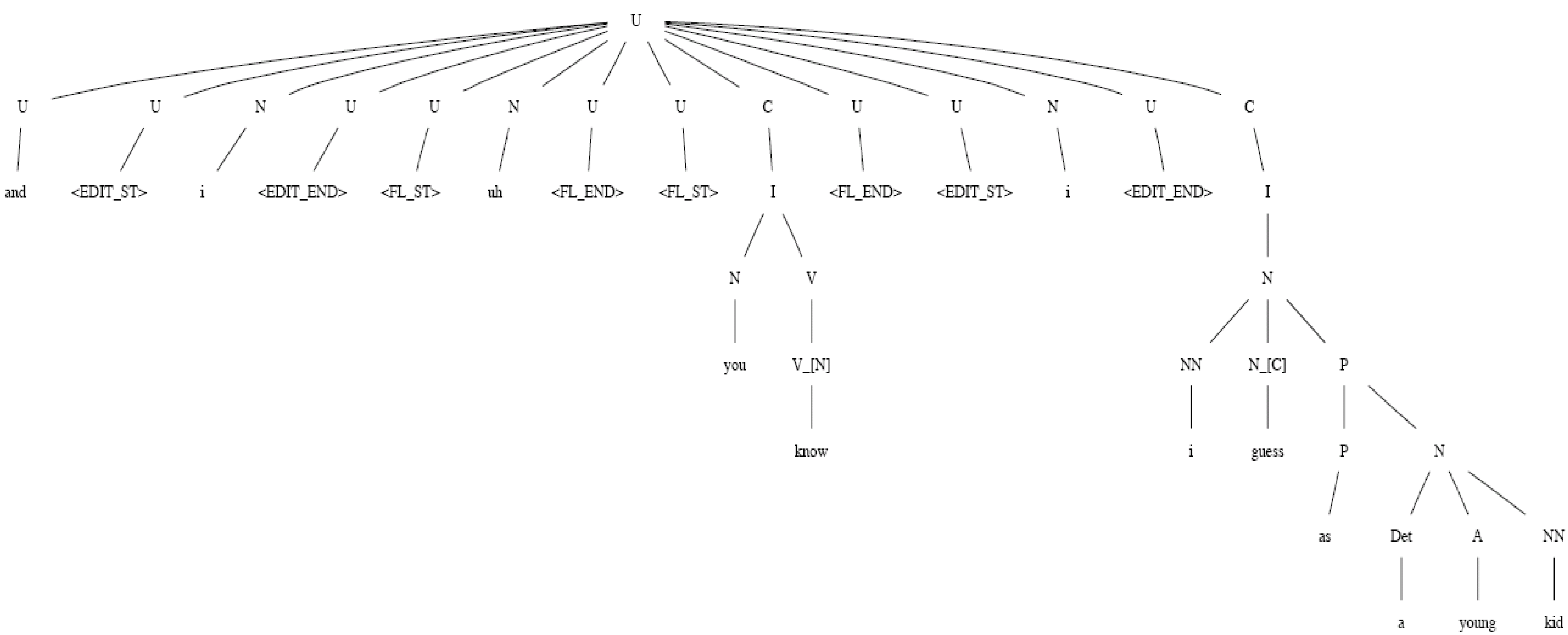
ENRICHED INPUT: …and <EDIT_ST> I <EDIT_END> <FL_ST> uh <FL_END> <FL_ST> you know <FL_END> <EDIT_ST> I <EDIT_END> I guess as a young kid…

Minipar: What is it?

- Minimalist approach to parsing (Dekang Lin, 1999)
  - Not a standard CKY of PCFG approach
  - Message passing design
- Design characteristics
  - Simplicity of grammar design
  - Efficiency: Produce structure that takes least effort to generate
- Two basic operations in minimalist theories: MERGE and MOVE
  - MERGE induced through feature Percolation/Checking.
  - MOVE induced through binding displaced element to trace.
- Advantages:
  - Parser computation is monotonic
  - Grammatical principles fall out from design
  - No training required
    - Can be applied directly to marked-up (MDE) input
    - Can test impact of meta-data on parsing directly
    - Caveat: Scores are lower (as expected) since it is untrained
- **MINI-BJD: Transforms Minipar to Treebank style**

Minipar: How can it be used?

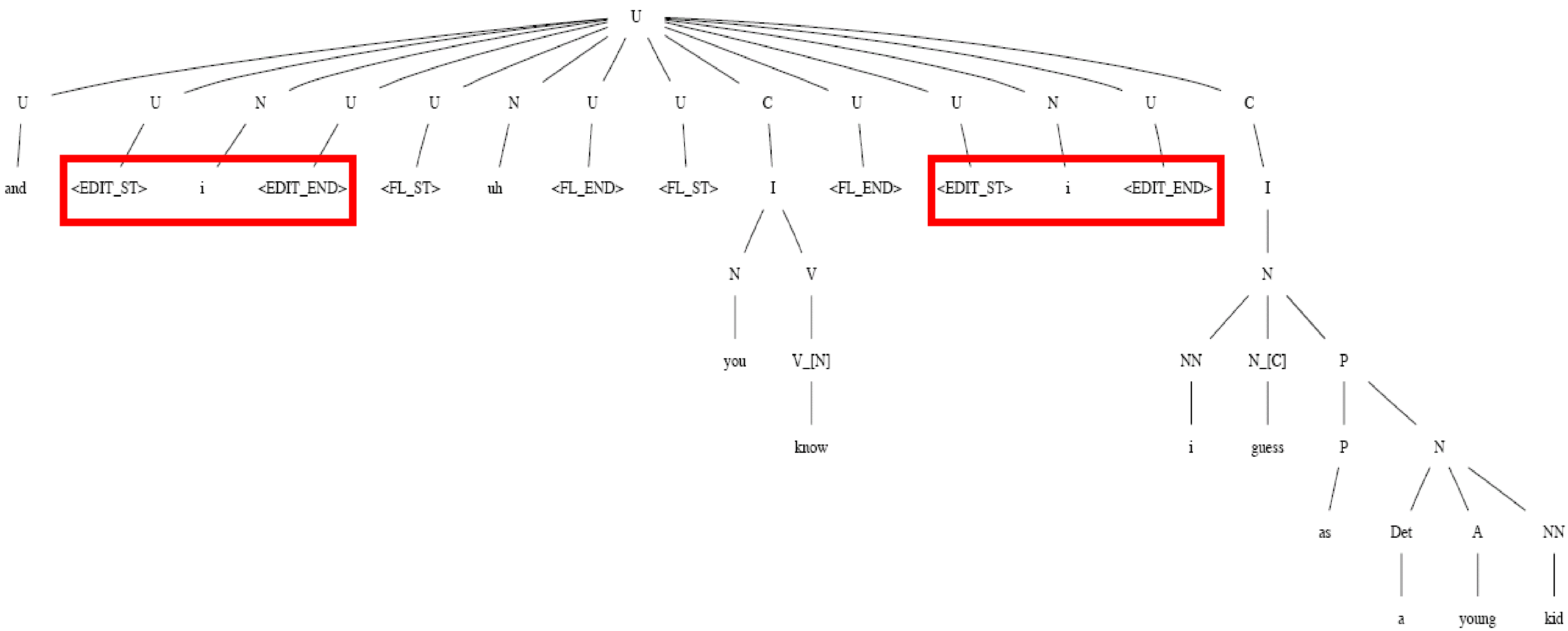
INPUT: …and I uh
you know I I guess as a young
kid …



Minipar applied directly to enriched string

Minipar: How can it be used?

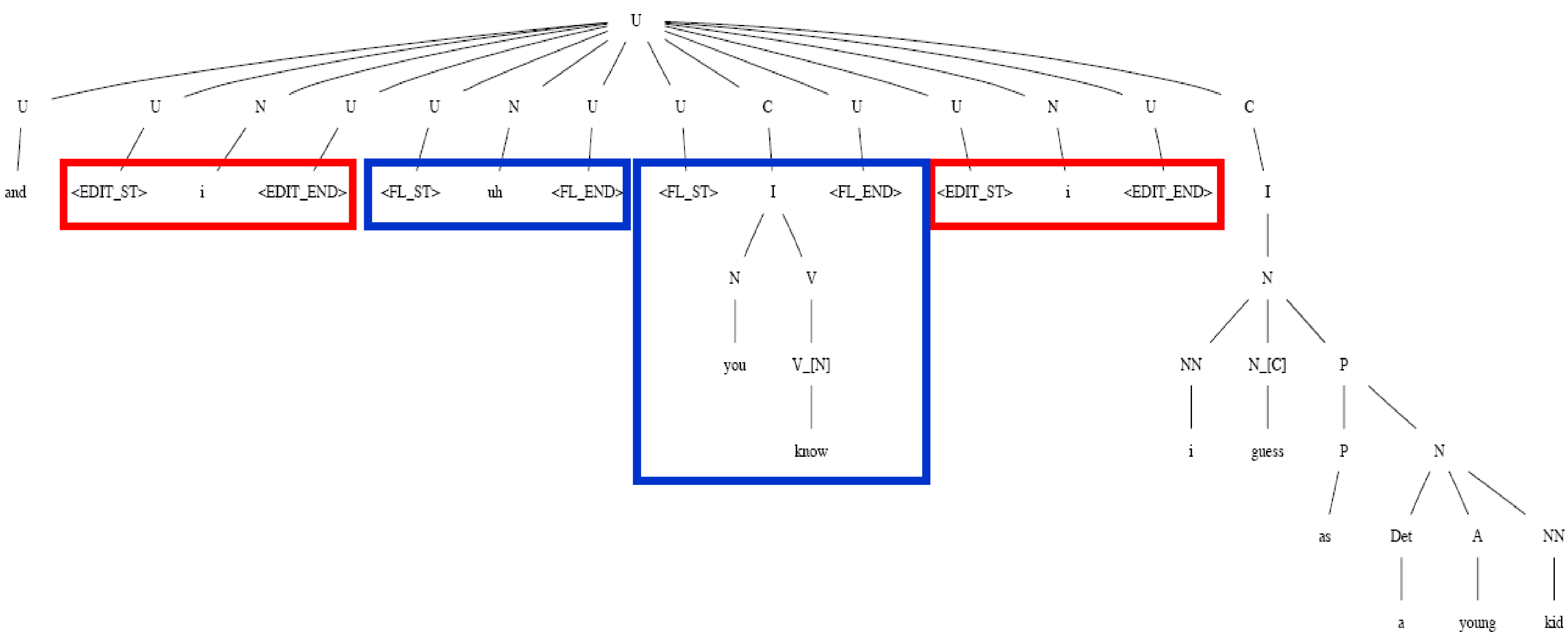
INPUT: …and <EDIT_ST> I <EDIT_END> uh you know <EDIT_ST> I <EDIT_END> I guess as a young kid …



Minipar applied directly to enriched string

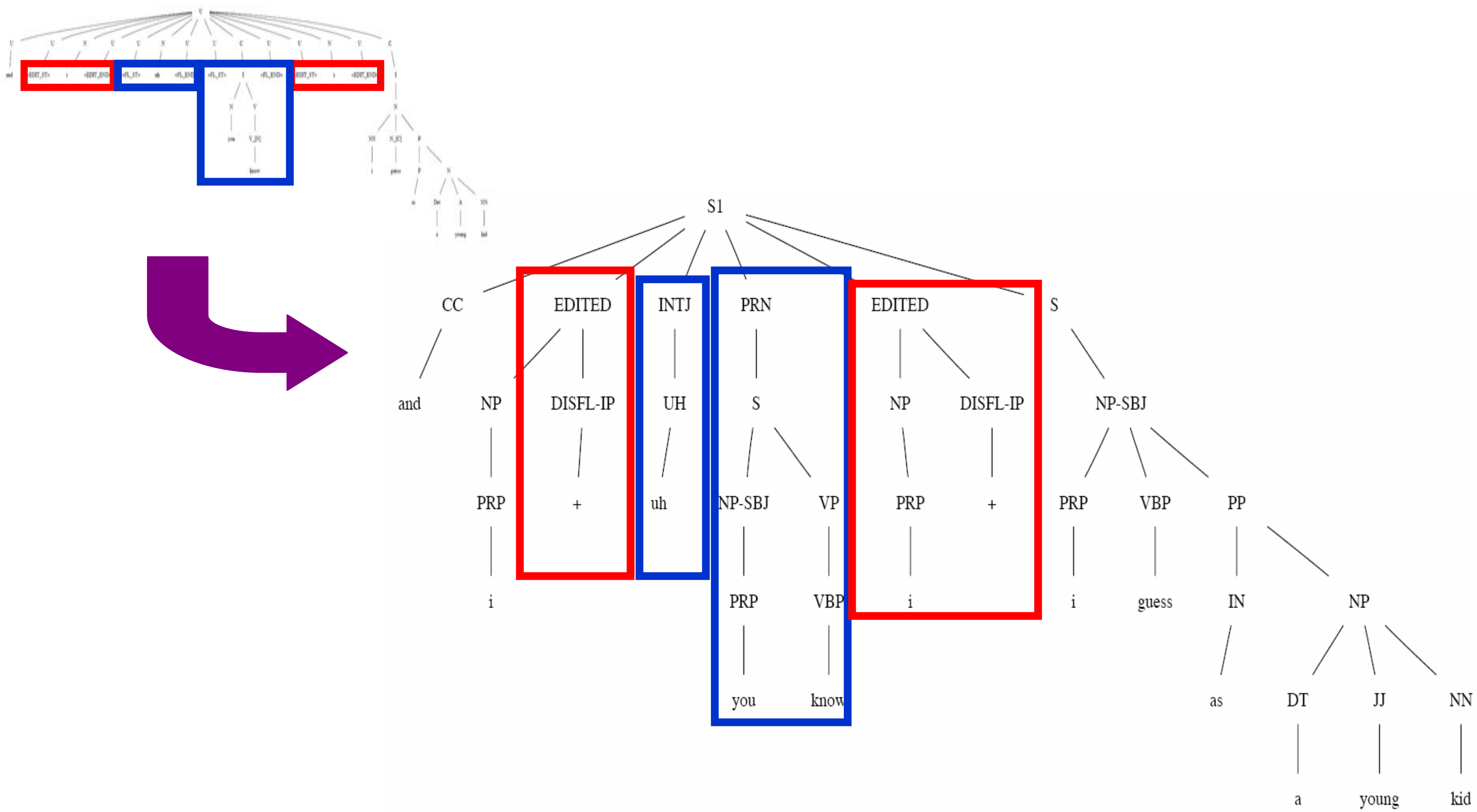
Minipar: How can it be used?

INPUT: …and <EDIT_ST> I <EDIT_END> <FL_ST> uh <FL_END> <FL_ST> you know <FL_END> <EDIT_ST> I <EDIT_END> I guess as a young kid …



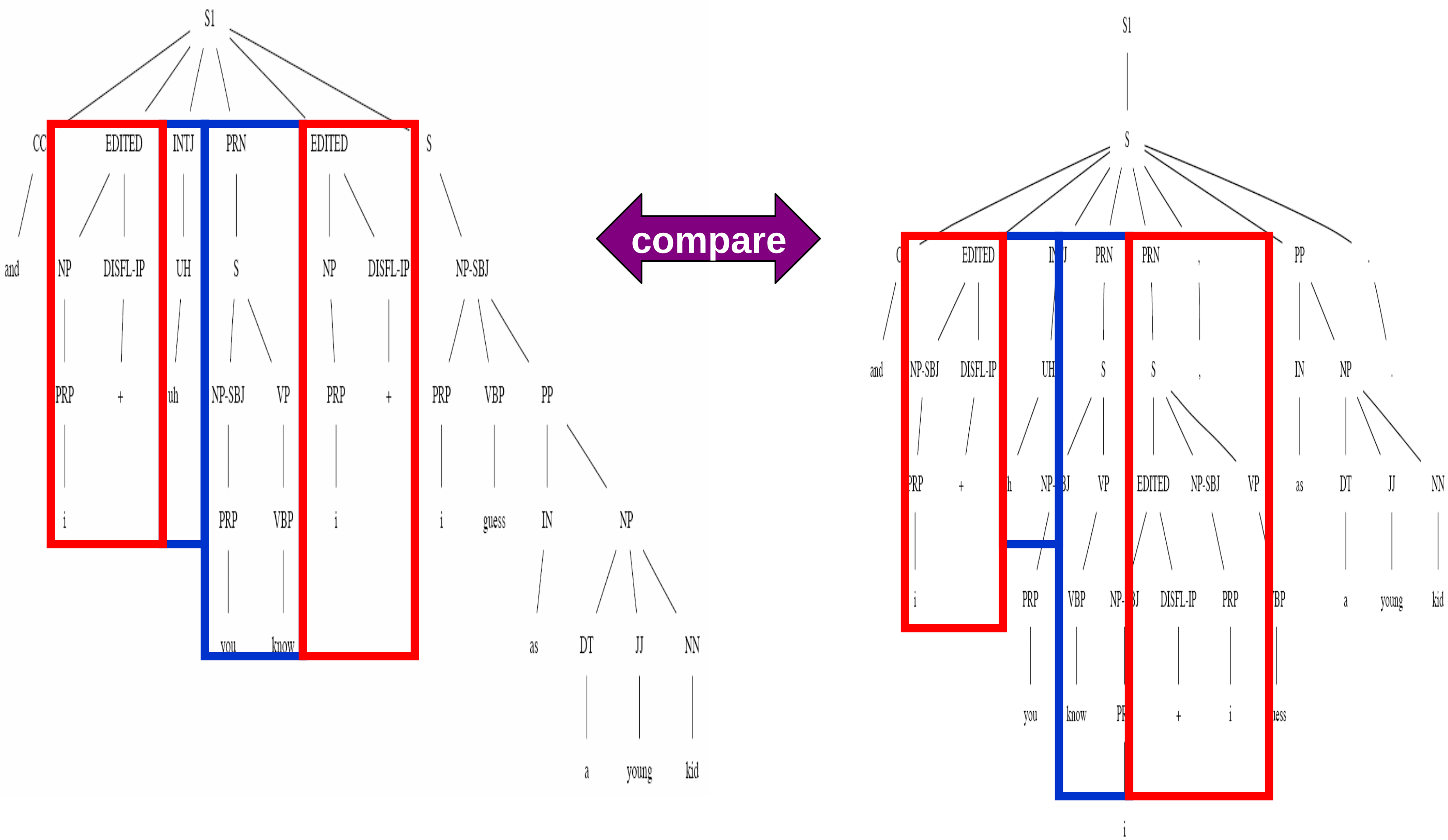
Minipar applied directly to enriched string

Minipar: How can it be used?



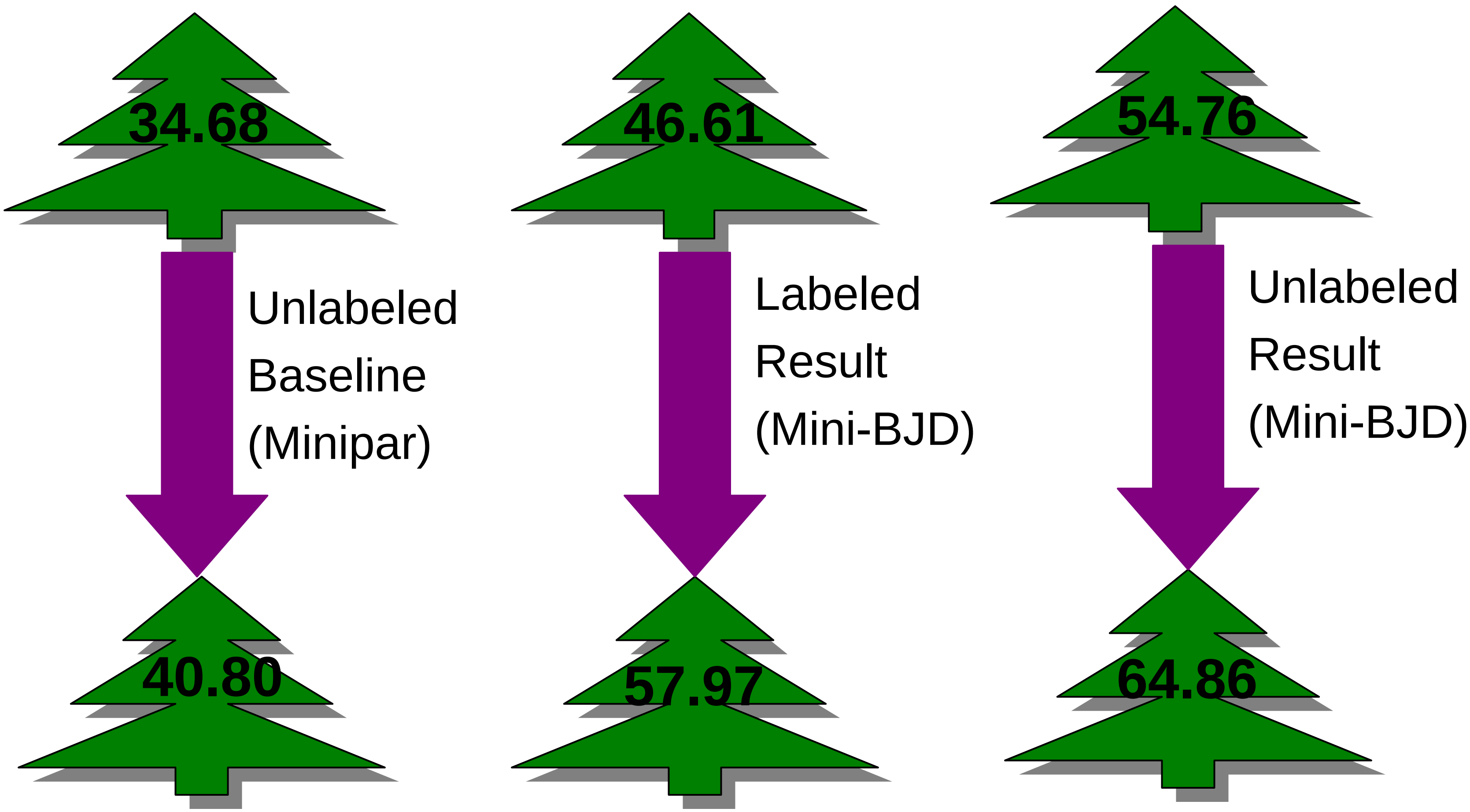
MINI-BJD transforms Minipar to Treebank Style

Minipar: How can it be used?



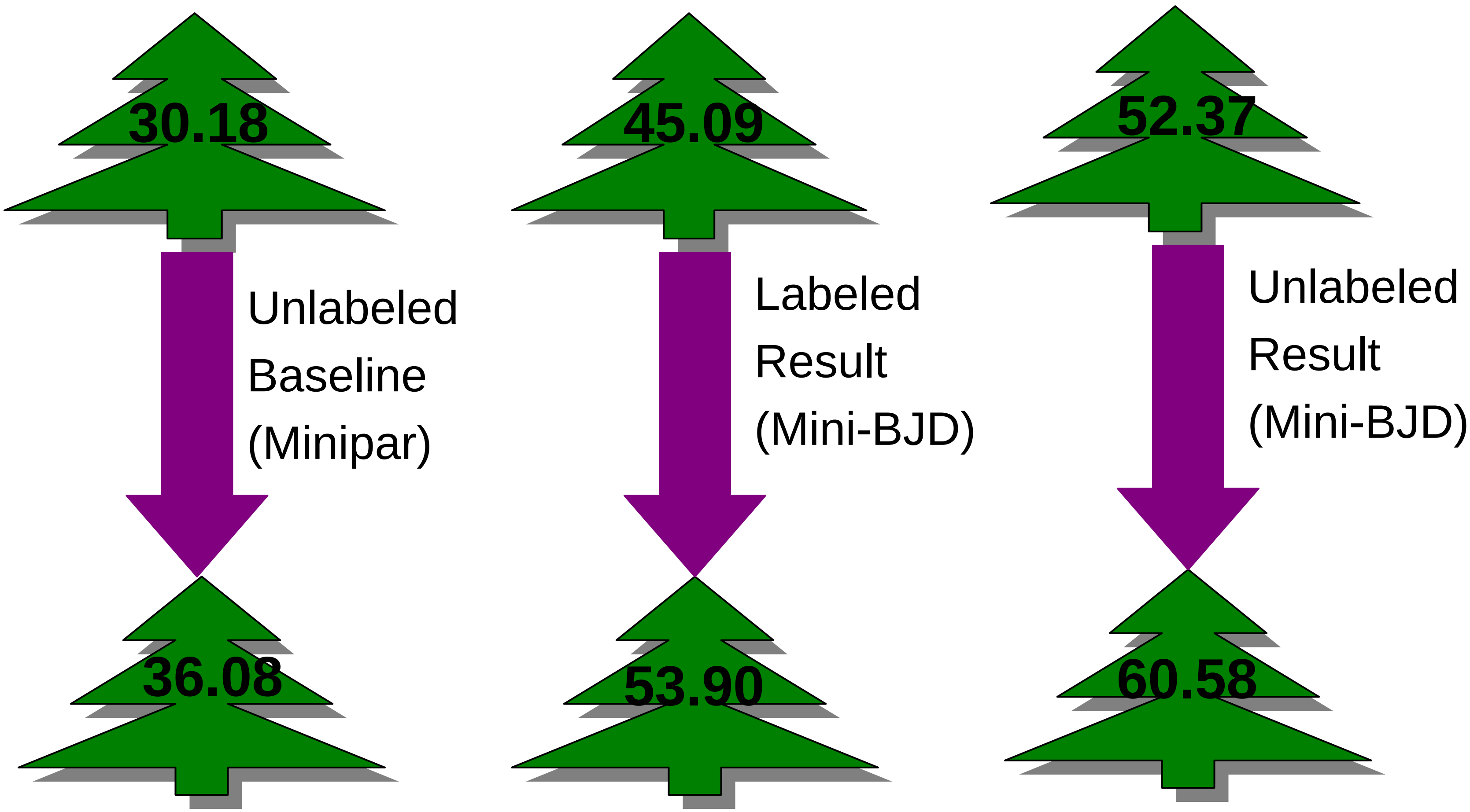
Evaluation: Compare MINI-BJD to Gold Standard

Point 1: Using Markup Instantly Improves Performance



Human-annotated metadata

Point 1: Using Markup Instantly Improves Performance



Machine-annotated metadata

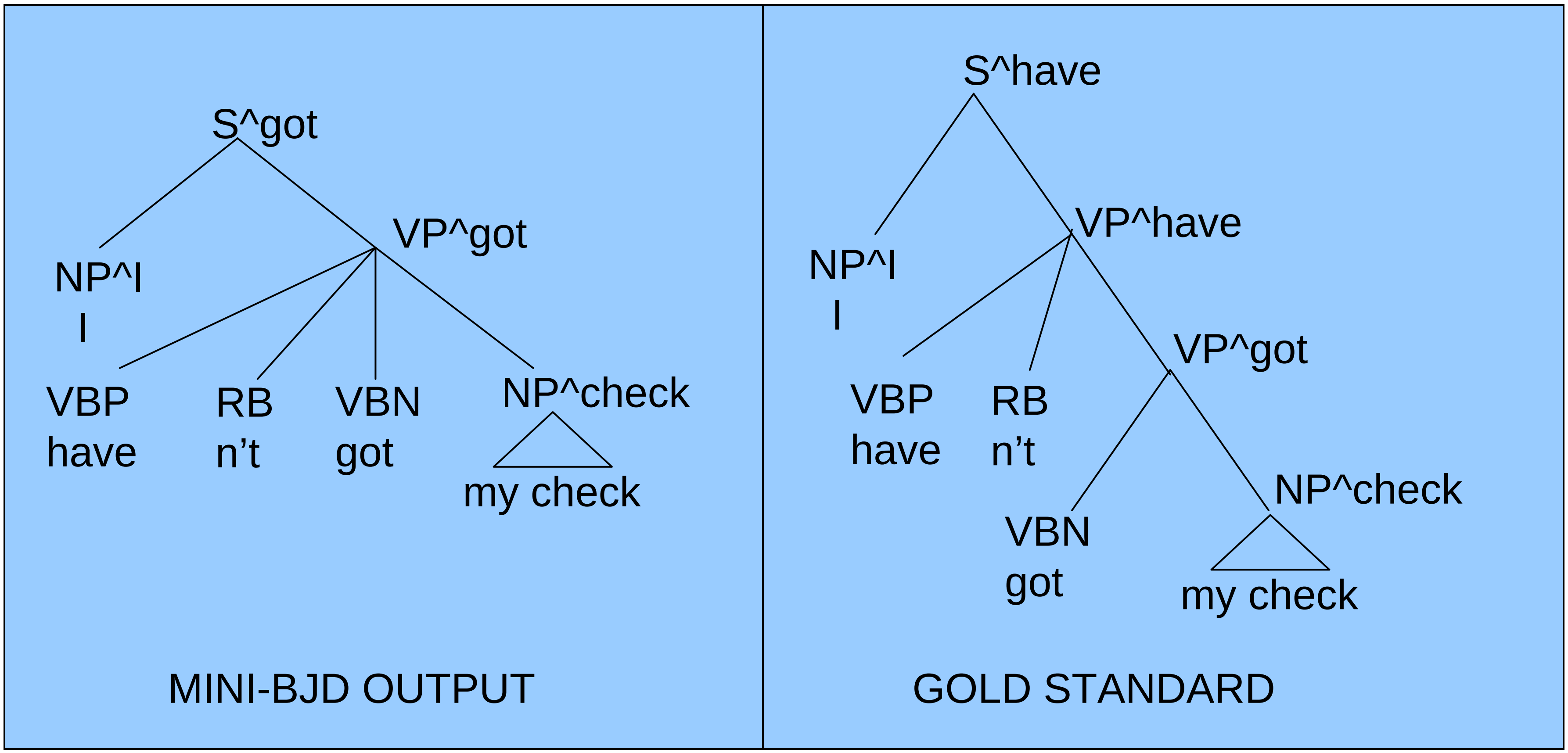
Point 1: Using Markup Instantly Improves Performance

	Human Transcriptions	**ASR Output**
Human Annotated Metadata	57.97 **(Up 11 pts)**	50.59 **(Up 9 pts)**
System Generated Metadata	53.90 **(Up 11 pts)**	47.08 **(Up 7 pts)**

Point 2: Head Percolation Tables Make a Difference!

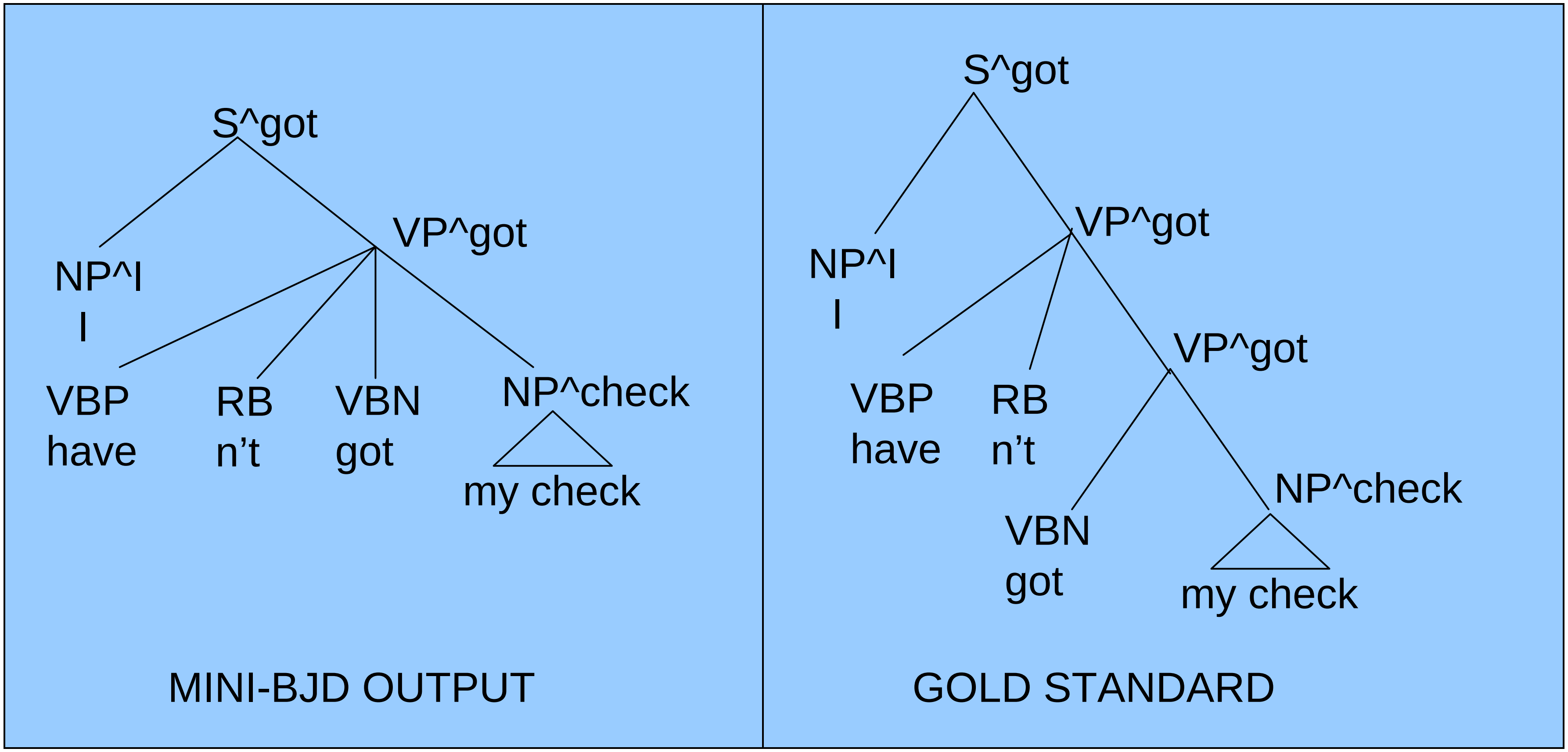
- Best MINI-BJD Parser Score
  - Labeled-Bracketing: 57.97
  - Head Dependency: **42.16** (Hwa), 40.65 (Charniak), 40.48 (Collins)
- Best Charniak Parser Score:
  - Labeled-Bracketing: 88.06
  - Head Dependency: 84.39 (Hwa), **85.68** (Charniak), 85.47 (Collins)
- What gives?
  - Hwa's tables expect short, fat trees: Geared toward characterizing appropriate dependency trees, e.g., GO head of "TO GO"
  - Charniak/Collins' tables expect tall, thin trees: Geared toward evaluation of syntactic trees, e.g., TO head of "TO GO".

Point 2: Head Percolation Tables Make a Difference!



CHARNIAK HEAD PERCOLATION F-SCORE: 0.375

Point 2: Head Percolation Tables Make a Difference!



HWA HEAD PERCOLATION F-SCORE: 0.75

Enrich Grammar

Adapt PCFG for Speech

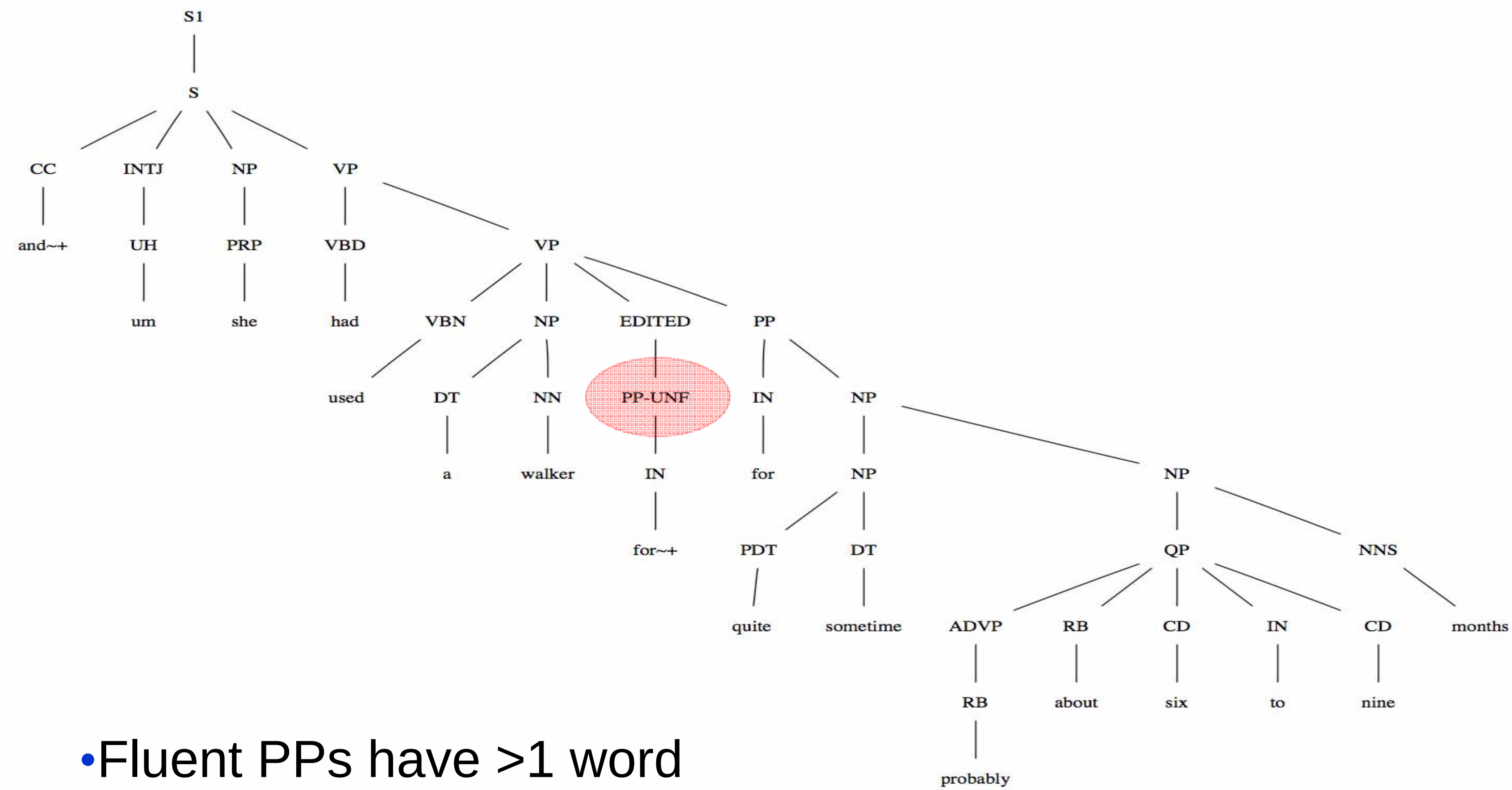
1. unfinished phrases
2. categories for reparanda

UNFinished phrases

- **this prepositional phrase is UNFinished:**

"and um she had used a walker [$_{PP}$ **for**] **for** quite sometime probably about six to nine months"

-UNF annotation



- Fluent PPs have >1 word
- LDC annotates lowest unfinished node

Better Viterbi parse with -UNF

	PARSEVAL F	EDIT-finding F
baseline	71.15	23.0
-UNF propagation	71.75	32.0

Train: Switchboard

Test: LDC Fisher “dev2” gold tags, gold sentence boundaries

Syntactic Parallelism

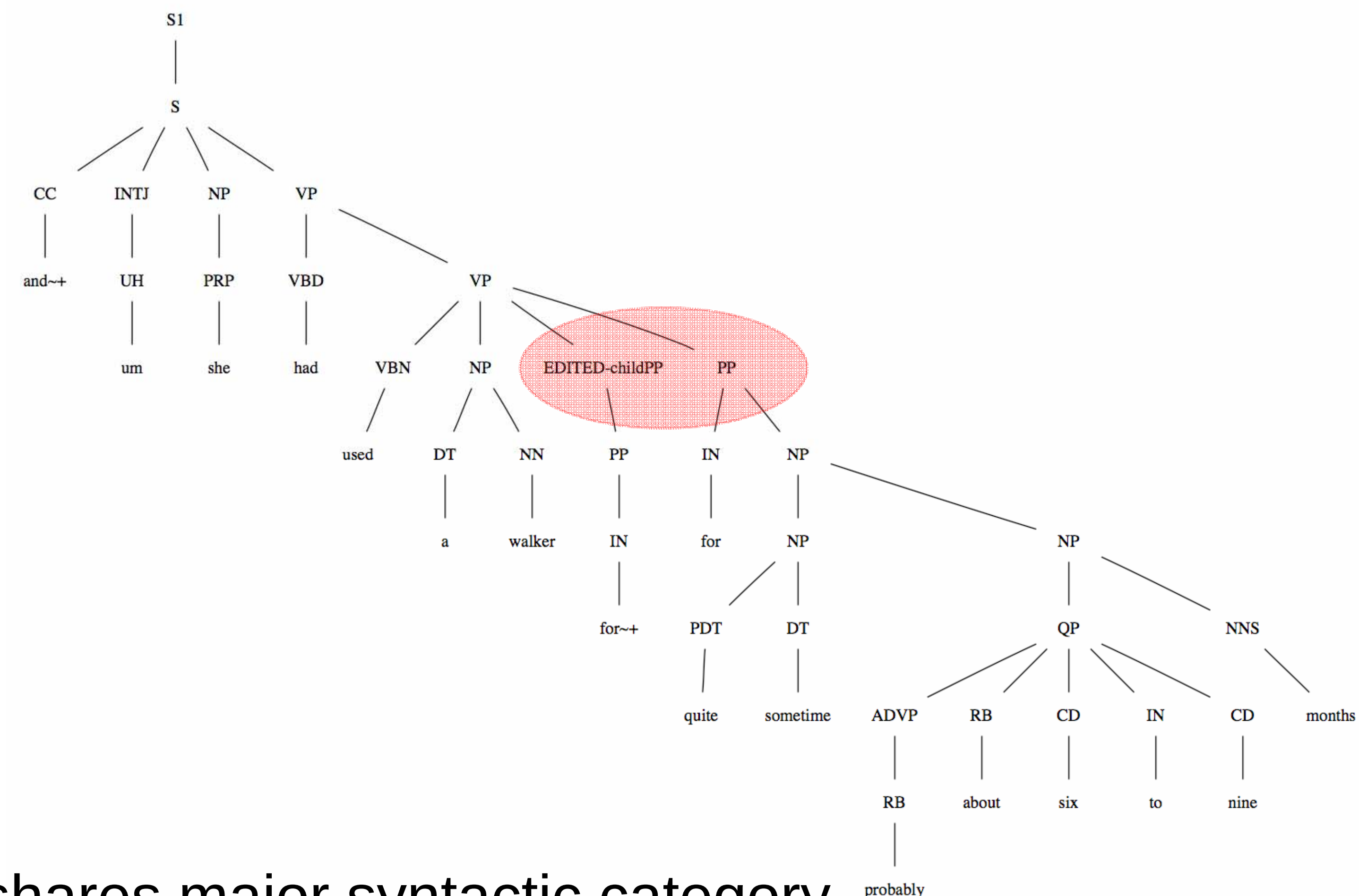
- **The unfinished prepositional phrase (PP) is parallel to a fluent PP**

"and um she had used a walker

[$_{\text{PP-UNF}}$ **for**] [$_{\text{PP}}$**for** quite sometime]

probably about six to nine months"

Parallel PPs



•repair shares major syntactic category

•capture with daughter annotation on EDITED

Better Viterbi parse with -childXP

	PARSEVAL F	EDIT-finding F
baseline	71.15	23.0
-UNF propagation	71.75	32.0
-child annotation	71.59	32.9

Independent Improvement

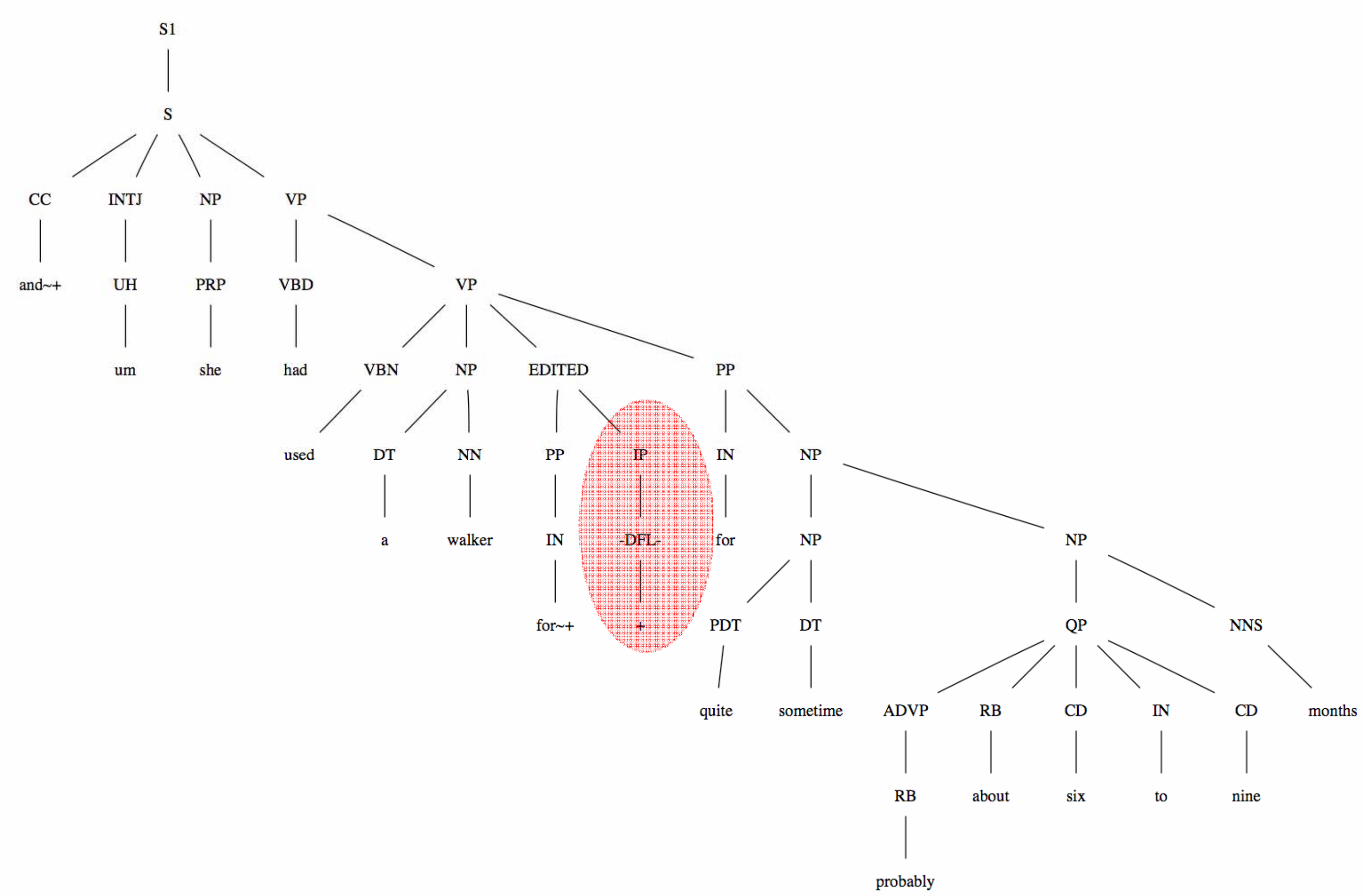
	PARSEVAL F	EDIT-finding F
baseline	71.15	23.0
-UNF propagation	71.75	32.0
-child annotation	71.59	32.9
both	72.45	42.3

Charniak: an improved EDIT-finder

	PARSEVAL F	EDIT-finding F
baseline	82.06	53.3
-UNF propagation	79.96	59.5
-child annotation	78.55	58.0
both	77.90	61.3

Charniak July 11 2005 non-reranking lexicalized parser
(parser performs tagging)

Where is the interruption point?



-UNF & -childXP synergize with IP

	PARSEVAL F	EDIT-finding F
oracle interruption point	75.84	81.7
oracle interruption point, -UNF & -childXP	76.53	87.9

Potential Benefit from ToBI mark

	PARSEVAL F	EDIT-finding F
baseline	67.66	21.5
“p” ToBI mark	64.89	30.6
“p” ToBI mark, -UNF, -childXP	64.29	34.1

reminder: “p” only signals interruption points 30% of the time

Parsers *can* adapt to speech

By enriching the given input string

- rewrite result

By enriching the given grammar

- create new rules

Parsers *can* adapt to speech

- By enriching the given input string
  - rewrite result

    (treating fillers as given)
- By enriching the given grammar
  - create new rules

    (ignored fillers)

Roadmap

- Background and Baseline Metadata Extraction
- Parsing Metrics and Impacting Factors
- Prosodic Structure
- Using Structural Knowledge to Improve Parsing
- Proposal: Disfluency and Parsing (Matt Lease)
- SU Reranking Experiments
- Proposal: Off-topic Detection (Robin Stewart)

Disfluencies and Parsing: English and Beyond

CLSP'05 Research Proposal

Matt Lease

Advisor: Eugene Charniak

JHU Workshop '05

Parsing and Spoken Structural Event Detection

BROWN

Disfluency in Baltimore Tourism corpus

Baltimore is the greatest city in America

Disfluency in Baltimore Tourism corpus

*Baltimore is the greatest city in [Maryland] * uh I mean America*

While **filled pauses** such as uh are easy to detect, **discourse markers** are far more frequent and often introduce ambiguity, requiring prosodic/contextual information for correct resolution

Did you know I do that?	–vs–	Did you know I do that?
I mean I do that.	–vs–	I mean I do that.
Is it like that one?	–vs–	Is it like that one?
I know well, I think.	–vs–	I know well I think…

Speech repairs hurt parse accuracy

- Cross-serial dependencies of repairs cause collateral damage to parse (Charniak and Johnson ’01)
- Interruption points modelled like punctuation help parsing in presence of repairs (Kahn ’05)
- Workshop results confirm these findings

Fillers also hurt parse accuracy

- Presence of INTJ and PRN reduces parse accuracy comparably to repairs (Engel et al. '02)
- A simple experiment using new MDE annotations
  - Given a parse tree, label each terminal as a filler iff. it's below an INTJ or PRN and commonly occurs as a filler
  - Using gold trees: 11.6% NIST error
  - Using best parser output: 23.7% NIST error
  - Conclusion: parser often misanalyzes fillers
  - As with repairs, these mistakes likely produce collateral damage to neighboring constituents as well

Disfluencies hurt Levantine parsing

Parsing Arabic Dialects team reports 21% F-score improvement using oracle disfluency detection

Details: Levantine transcripts, Chiang parser trained on Penn MSA treebank, gold POS tags, from deleting: repairs, unfinisheds, interjections, and filled pauses from test data, F=63% vs. F=42%

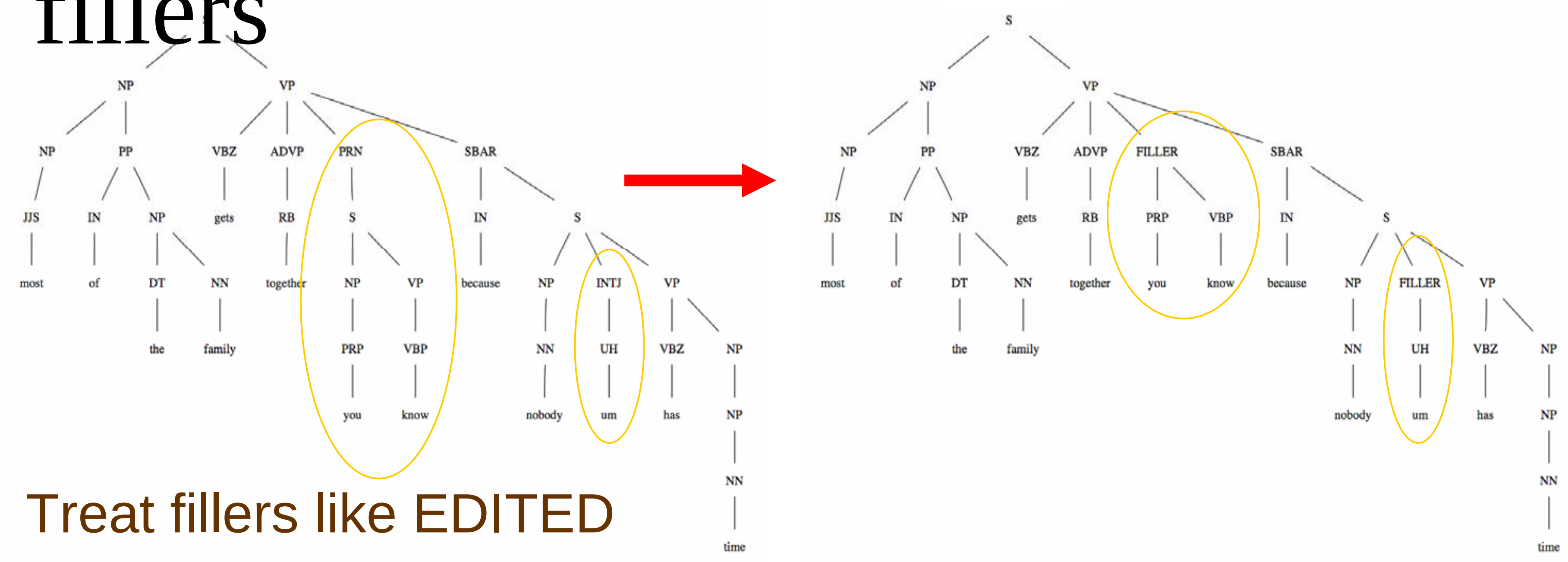
Proposed Work

- Analyze and reconcile guidelines for filler annotation
- Investigate alternative syntactic filler representation
- Explore noisy channel-based disfluency modelling
- Cross-linguistic study of syntax-disfluency interaction

What is a filler, really?

- Treebank and SimpleMDE guidelines define fillers independently of one other
- A unified definition would benefit the community, both scientifically and in the creation of consistent resources
- Preliminary analysis suggests resolution is possible
- We have made initial proposal of revised treebanking guidelines to LDC, but more careful analysis needed

A new syntactic representation of fillers



- Treat fillers like EDITED
  - Transform trees: prune fillers and reinsert each filler span under a new, flat FILLER constituent (non-filler INTJ, PRN left unchanged)
  - Measure oracle vs. syntax-driven detection
  - Use relaxed parseval and treat FILLER like EDITED (effectively combine into single NON-SEMANTIC category)

Noisy-channel modelling of disfluency

$$\hat{f} = \arg\max_{S} P(F \mid D) = \arg\max_{S} P(F)P(D \mid F)$$

- Idea: recover most-likely fluent utterance underlying given observed, possibly disfluent utterance (Honal & Schultz, 2003)

- Directions

  - Automatically learn parser mistake patterns correlated with disfluency using text-based *compression* noisy-channel model (Knight & Marcu, 2000)

  - Investigate bootstrapped repair detection on unannotated or partially annotated corpora (e.g. SimpleMDE does not annotate end of speech repair)

  - Incorporate prosody (very limited use to date using noisy-channel framework)

Disfluency and parsing: Levantine and Mandarin

- *Levantine data: pilot MDE corpus* NEW! *and CallHome treebank of conversational speech* NEW!
- *Mandarin data: pilot MDE corpus* NEW! *and CallHome transcripts (with limited filler annotations)*
- Idea: exploit newly available data to study interaction between syntax and disfluency, applying models shown to be effective in English

Proposed Work

- Analyze and reconcile guidelines for filler annotation
- Investigate alternative syntactic filler representation
- Explore noisy channel-based disfluency modelling
- Cross-linguistic study of syntax-disfluency interaction

Thanks!

Roadmap

- Background and Baseline Metadata Extraction
- Parsing Metrics and Impacting Factors
- Prosodic Structure
- Using Structural Knowledge to Improve Parsing
- Proposal: Disfluency and Parsing (Matt Lease)
- SU Reranking Experiments
- Proposal: Off-topic Detection (Robin Stewart)

SU Detection trials

N-best reranking: significant improvements over strong baseline

- effective candidate extraction
- feature extraction
  - multiple parsers providing syntactic features
  - prosodic, conversation turn, and lexical features
- STT versus reference transcripts
- Parameter estimation
  - SU accuracy
  - parsing accuracy

SU detection in n-best scenario

- Conversation side is a very long sequence
  - Average length in dev set $>$ 500 words; max over 1000
- Every word boundary is a potential segmentation point
- Oracle accuracy of 1000-best list over conversation side not much better than 1-best accuracy
- Need a better method for effective reranking
  - Will enable us to include features inaccessible to the finite-state sequence model baseline
- Will re-rank over relatively small pieces of the conversation side

Accuracy of baseline on Dev set

Baseline Posterior	Percent Accurate	Percent of Word Boundaries
$\boldsymbol{x > 0.95}$	97.9	8.2
$\boldsymbol{x < 0.05}$	99.4	77.0
$\boldsymbol{0.05 \leq x \leq 0.95}$	78.1	14.8

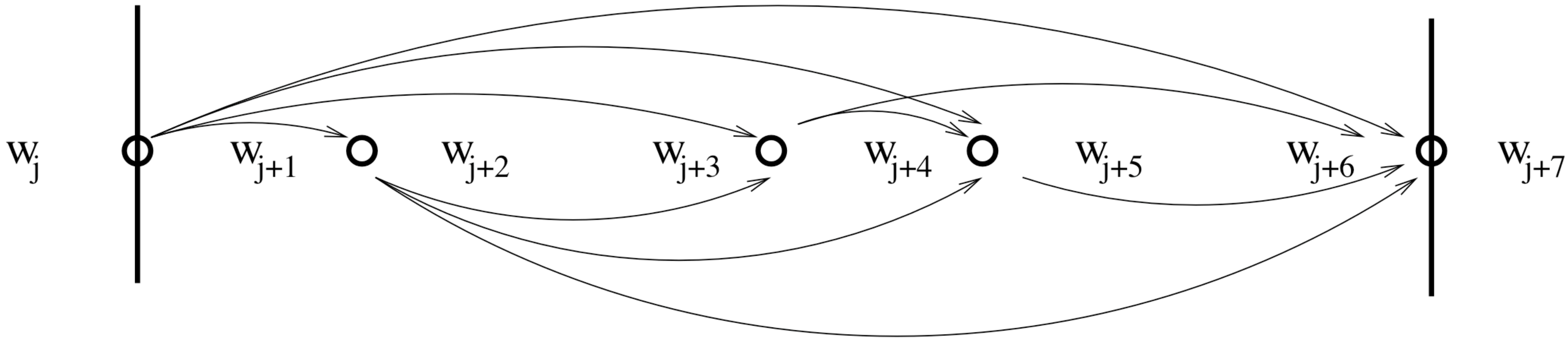
Begin establishing candidates by

- fixing all very high posterior points as SU boundaries
- fixing all very low posterior points as non-boundaries

N-best extraction

- Two-stage n-best candidate selection, using baseline model
- First stage: establish "fields" over which candidates will be ranked
  - Segment at all word boundaries with baseline probability of segmentation over parameter p
  - Choose highest probability internal word boundary to segment "fields" with more than k words
- Second-stage: create candidates for each field
  - Choose the j highest probability word boundaries within the field as hypothesized segmentations
  - Do not hypothesize a segment if the probability is below q

Picture



A particular run is noted as $\boldsymbol{p}$-$\boldsymbol{k}$-$\boldsymbol{j}$-$\boldsymbol{q}$, e.g. 95-50-10-05, meaning:

- Segment at all word boundaries with probability $\geq \mathbf{0.95}$
- Keep segmenting until all "fields" of length $\leq \mathbf{50}$
- Put up to $\mathbf{10}$ internal hypothesis points, if possible
- Don't hypothesize points with probability $\leq \mathbf{0.05}$

$\longrightarrow$ 95-50-10-05 gives us 97.4 oracle accuracy on Dev2, tractable candidate sets

N-best reranking

- Once we have candidates, we can extract features from candidates for use in a reranker
  - e.g. run a parser on segments, derive features from parses
- We have been using Mark Johnson's MaxEnt reranker
  - Optimizes a regularized globally conditioned log likelihood
  - Used for parse reranking in Charniak and Johnson (ACL, 2005)
- Have code to combine heterogeneous features into single model
- Features derived from the candidate; from individual segments within the candidate; or words

Reranking for SU accuracy

	Candidate	SU accuracy
	1	0.8
→	2	0.9
	3	0.8
→	4	0.9
	5	0.6
	6	0.7
	7	0.5

Empirical setup

- Dev1 set 75,000 words
- Dev2 set 35,000 words
- Eval set 35,000 words
- Baseline SWBD train set 400,000 words (no STT)
  - Found that gains were higher on Dev2 when training on Dev1 rather than the Baseline training set
  - Since wanted STT and REF trials, all reported training on Dev1
- Reference transcript and STT conditions

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

Empirical Results (Dev2 reference transcript)

System	Ftr. Set	No. of Features	F-measure Accuracy	NIST Error	Train Time	Max Mem.
Baseline	1	1	84.9	29.4	-	-
Rerank	1-8	163	85.9	28.1	12.3s	80MB
Rerank	1-12	22435	86.0	27.8	30.9s	80MB
Rerank	1-15	183837	86.8	26.4	678.8s	1.3GB

Features:

1) posterior from baseline
2) number of field internal segments guessed
3) max/min segment lengths
4) average segment length
5) n-gram score
6) Charniak parser LM score
7) root symbol of viterbi trees
8) root symbol + number of children of viterbi trees
9) non-root symbols of viterbi trees
10) non-root symbols + no. of children of viterbi trees
11) Initial and final unigrams/bigrams
12) Initial and final unitags/bitags
13) Speaker change/backchannel indicators
14) Baseline annotated disfluency information
15) Constraint-Dependency Grammar (CDG) Parser-derived features

Reranking with STT transcripts

- To do this reranking, we needed reference SU boundaries imposed upon the STT transcript
- Producing this is not straightforward
  - Some SU boundaries correspond to locations with no word boundary in the STT transcript, hence must be omitted
  - Resulting “gold” SU boundaries have a 5.4% NIST error
- Hence re-ranking with this objective is less effective for SU detection than in the reference case
- Further, small training set size hurts more for noisy STT output

Empirical Results (Dev2 STT)

System	Ftr. Set	No. of Features	F-measure Accuracy	NIST Error	Train Time	Max Mem.
Baseline	1	1	80.4	37.9	-	-
Rerank	1-5	88	81.0	36.5	4.3s	80MB
Rerank	1-10	22094	81.2	36.3	17.0s	144MB
Rerank	1-13	175047	81.3	36.1	560s	1.4GB

Features:

1) posterior from baseline
2) n-gram score
3) Charniak parser LM score
4) root symbol of viterbi trees
5) root symbol + number of children of viterbi trees
6) non-root symbols of viterbi trees
7) non-root symbols + no. of children of viterbi trees
8) Initial and final unigrams/bigrams
9) Initial and final unitags/bitags
10) TOBI based prosodic labels
11) Speaker change/backchannel indicators
12) Baseline annotated disfluency info
13) Constraint-Dependency Grammar (CDG) Parser-derived features

Empirical Results (Eval)

System	F-measure Accuracy	NIST Error
Baseline REF	84.9	28.9
Rerank REF	86.3	26.9

Baseline STT	80.0	38.3
Rerank STT	80.4	37.4

REF result significant at $p < 0.0005$

STT result not statistically significant

Reranking paradigm

- One great benefit of the reranking paradigm is the ability to focus on other objectives
- SU boundary detection is of utility for downstream processing
  - Formatting for ease of reading
  - NLP annotations such as parsing
  - Also for subsequent machine translation
- Very straightforward to modify this approach to serve a downstream objective

Reranking for SU accuracy

	Candidate	SU accuracy	Parsing accuracy
	1	0.8	0.7
$\longrightarrow$	2	0.9	0.7
	3	0.8	0.8
$\longrightarrow$	4	0.9	0.8
	5	0.6	0.7
	6	0.7	0.6
	7	0.5	0.5

Reranking for parsing accuracy

	Candidate	SU accuracy	Parsing accuracy
	1	0.8	0.7
	2	0.9	0.7
$\longrightarrow$	3	0.8	0.8
$\longrightarrow$	4	0.9	0.8
	5	0.6	0.7
	6	0.7	0.6
	7	0.5	0.5

Parse accuracy reranking (Dev set)

System	Optimized for	SU performance P	R	F	NIST	Bracketing F-measure	H-Dep F-measure
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

Summary of SU reranking

- Significant system improvements using very small training sets
- Need further work on features for STT case
  - More untried dependency-based features
  - More untried prosodic+syntactic features
- Will soon produce results combining Dev1 and Dev2 as training
- Ability to optimize for other objectives is an interesting direction
  - Also try different balance between precision and recall
- Would be nice to have STT and/or parsing n-best included in optimization

Roadmap

- Background and Baseline Metadata Extraction
- Parsing Metrics and Impacting Factors
- Prosodic Structure
- Using Structural Knowledge to Improve Parsing
- <u>Proposal:</u> Disfluency and Parsing (Matt Lease)
- SU Reranking Experiments
- <u>Proposal:</u> Off-topic Detection (Robin Stewart)

Post-Workshop Research Proposal

Off-Topic Detection:

Metaconversation and Small Talk

Robin Stewart (Williams College)

Supervisor: Yang Liu (ICSI & UT-Dallas)

Facilitator: Andrea Danyluk (Williams College)

Example

(Topic: Personal Habits)

...
R: Uh, I'm in college so, like, my drinking is pretty cheap.
Maybe like five bucks a week.
L: Oh, that's not bad.
R: [LAUGH] Yeah, it's pretty cheap.
L: Mhm.
Wait, what college do you go to by the way?
R: University of Illinois.
L: Really, in Champagne?
R: Yeah. In Champagne.
L: Oh, wow.
R: And you live in New York?
L: Yeah.
R: Interesting.
L: Yeah.
But - um - So anyways I guess we're off topic again [LAUGH].
R: [LAUGH] Yeah
L: Um- [LAUGH] um, what were the other things on the list?
Oh yeah, overeating.
See, you know what I heard about, um, overeating is that - or - or just in general, like, you know, obesity and everything is that - um -
Right now smoking is the number one cause of death in the country.
But then pretty soon it's going at - um - switch over to obesity.
R: Yeah. I've - I've heard about that too.

Example

(Topic: Personal Habits)

...

R: Uh, I'm in college so, like, my drinking is pretty cheap.
Maybe like five bucks a week.
L: Oh, that's not bad.
R: [LAUGH] Yeah, it's pretty cheap.
L: Mhm.
Wait, what college do you go to by the way?
R: University of Illinois.
L: Really, in Champagne?
R: Yeah. In Champagne.
L: Oh, wow.
R: And you live in New York?
L: Yeah.
R: Interesting.
L: Yeah.
But - um - So anyways I guess we're off topic again [LAUGH].
R: [LAUGH] Yeah
L: Um- [LAUGH] um, what were the other things on the list?
Oh yeah, overeating.
See, you know what I heard about, um, overeating is that - or - or just in general, like, you know, obesity and everything is that - um -
Right now smoking is the number one cause of death in the country.
But then pretty soon it's going at - um - switch over to obesity.
R: Yeah. I've - I've heard about that too.

On topic

Example

(Topic: Personal Habits)

...

R: Uh, I'm in college so, like, my drinking is pretty cheap.
Maybe like five bucks a week.
L: Oh, that's not bad.
R: [LAUGH] Yeah, it's pretty cheap.
L: Mhm.
Wait, what college do you go to by the way?
R: University of Illinois.
L: Really, in Champagne?
R: Yeah. In Champagne.
L: Oh, wow.
R: And you live in New York?
L: Yeah.
R: Interesting.
L: Yeah.
But - um - So anyways I guess we're off topic again [LAUGH].
R: [LAUGH] Yeah
L: Um- [LAUGH] um, what were the other things on the list?
Oh yeah, overeating.
See, you know what I heard about, um, overeating is that - or - or just in general, like, you know, obesity and everything is that - um -
Right now smoking is the number one cause of death in the country.
But then pretty soon it's going at - um - switch over to obesity.
R: Yeah. I've - I've heard about that too.

On topic

Small talk

Example

(Topic: Personal Habits)

...
R: Uh, I’m in college so, like, my drinking is pretty cheap.
Maybe like five bucks a week.
L: Oh, that’s not bad.
R: [LAUGH] Yeah, it’s pretty cheap.
L: Mhm.
Wait, what college do you go to by the way?
R: University of Illinois.
L: Really, in Champagne?
R: Yeah. In Champagne.
L: Oh, wow.
R: And you live in New York?
L: Yeah.
R: Interesting.
L: Yeah.
But - um - So anyways I guess we're off topic again [LAUGH].
R: [LAUGH] Yeah
L: Um- [LAUGH] um, what were the other things on the list?
Oh yeah, overeating.
See, you know what I heard about, um, overeating is that - or - or just in general, like, you know, obesity and everything is that - um -
Right now smoking is the number one cause of death in the country.
But then pretty soon it's going at - um - switch over to obesity.
R: Yeah. I've - I've heard about that too.

On topic

Small talk

Meta-conversation

Example

(Topic: Personal Habits)

...

R: Uh, I'm in college so, like, my drinking is pretty cheap.
Maybe like five bucks a week.
L: Oh, that's not bad.
R: [LAUGH] Yeah, it's pretty cheap.
L: Mhm.
Wait, what college do you go to by the way?
R: University of Illinois.
L: Really, in Champagne?
R: Yeah. In Champagne.
L: Oh, wow.
R: And you live in New York?
L: Yeah.
R: Interesting.
L: Yeah.
But - um - So anyways I guess we're off topic again [LAUGH].
R: [LAUGH] Yeah
L: Um- [LAUGH] um, what were the other things on the list?
Oh yeah, overeating.
See, you know what I heard about, um, overeating is that - or - or just in general, like, you know, obesity and everything is that - um -
Right now smoking is the number one cause of death in the country.
But then pretty soon it's going at - um - switch over to obesity.
R: Yeah. I've - I've heard about that too.

On topic

Small talk

Meta-conversation

Definitions

- **Small Talk:** Conversation that is not related to or not contributing to the assigned topic.
- **Metaconversation:** Conversation about the assigned topic, the task, and the phone call.
- **On-Topic:** Everything else.

Definitions

- **Small Talk:** Conversation that is not related to or not contributing to the assigned topic.
- **Metaconversation:** Conversation about the assigned topic, the task, and the phone call.
- **On-Topic:** Everything else.

Goal: Automatically classify sentences in recorded telephone conversations

Motivations

- Just as "edit" regions can be removed to improve parsing, "small talk" regions could be removed to improve **information extraction**.
  (someone searching for weather information shouldn't get audio clips of "so, how's the weather?")

- Both metaconversation and small talk regions may help to identify changes in topic for **new topic detection**.
  - Meta: "Now we're supposed to talk about US public schools..."
  - Small talk: fills the gap between more-significant topics

Motivations

- Can also be applied to:
  - Meeting corpora
    ("You should have seen the traffic today...")
    ("Let's talk about the quarterly revenue report.")
  - Broadcast news
    ("I'm glad I'm safe inside the studio!")
    ("We now go live to Jim for an update.")
  - Surreptitiously recorded telephone conversations
    ("We had mac and cheese again tonight")
    ("So I was calling you because...")
  - Lectures, etc.

Related Work

- "Off-talk" detection for human-machine interaction (University of Munich)
  - "Oh, I have to click on that with the mouse"
- Social dialogue with conversational agents (Northwestern, MIT Media Lab)
  - Generating and responding to small talk with human users
- **NIST** Topic Detection and Tracking benchmark tasks (1998-2004)
  - Supervised and unsupervised classification techniques
  - Evaluation metrics

Proposal

- Weakly supervised classification of sentences
- Local classification techniques:
  - Naive Bayes (“bag of words”) classifier
  - Maximum-entropy (MaxEnt) classifier
  - Support Vector Machine (SVM) classifier
- Sequence decoding:
  - Hidden Markov Model (HMM)
  - Conditional Random Field (CRF)
- Train the classifier on a small set, use it to automatically “annotate” a much larger corpus, then iteratively re-train on the larger corpus

Feature Extraction

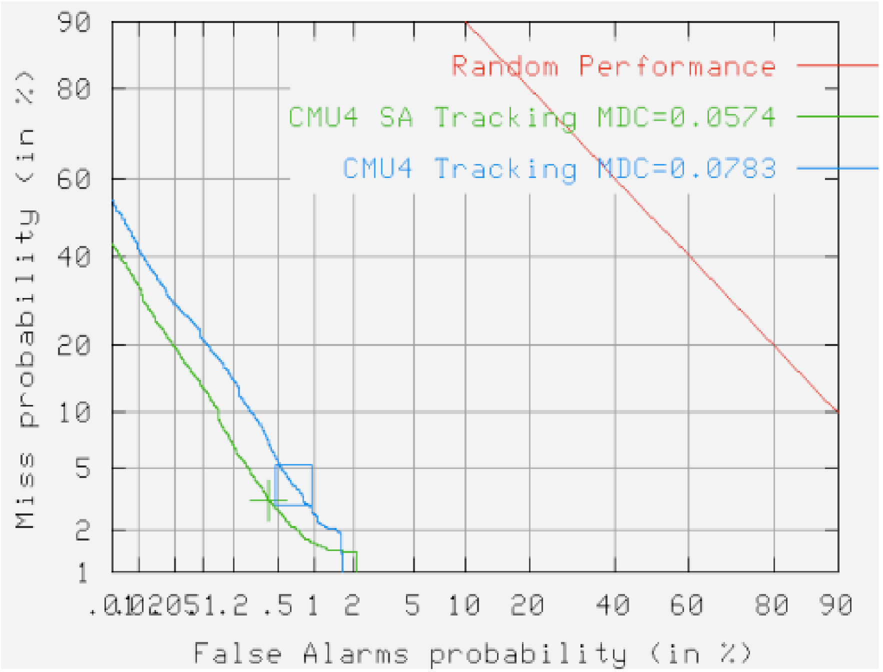
- Similar to our metadata reranking system
- Features which might prove useful:
  - Bigram or trigram language model
  - Key words such as filled pauses and discourse markers
  - Speaker changes and overlap
  - Duration of pauses
  - Frequency of awkward laughs
  - Etc.
- Easily extracted from our corpus

Annotation

- I've fully annotated 5 conversations, and looked over many others.
  - The time it takes to annotate is at *most* twice the length of the conversation.
  - We expect high annotator agreement.
- Weakly supervised learning techniques minimize the amount of annotation needed.
  - Need ~ 3 hours of training data (30 conversations) and another 3 hours for evaluation
  - 2 annotators for each conversation, plus a "tiebreaker"
  - ~ 30 hours of work = feasible
- Create annotation spec

Evaluation

- Accuracy - % of sentences correctly identified
- NIST metrics for Detection Evaluation
  - Detection Error Tradeoff curves
    - uses probability estimates to graph the tradeoff between misses and false alarms



We will find out:

- How well can off-topic regions be detected using standard machine learning techniques?
- How much training data is needed?
- Which machine learning algorithms work well?
- What features are effective?
- What is the effect of ASR and MDE errors?
- How well do ASR and MDE systems perform in on-topic vs. off-topic regions?

Conclusion

- **Useful**
  - Improve Information Extraction and New Topic Detection
- **Generalizable**
  - Meetings, Broadcast News, Phone Calls, ...
- **Feasible**
  - Builds on NIST TDT benchmark tasks
  - Small amount of annotation

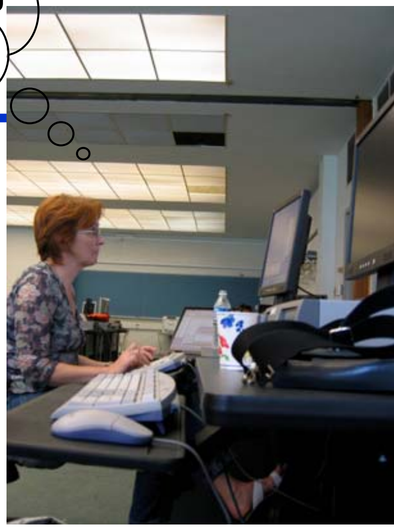
Acknowledgements

- **The team:** Bonnie Dorr, John Hale, Mary Harper, Anna Krasnyanskaya, Matt Lease, Yang Liu, Brian Roark, Zak Shafran, Matt Snover, Robin Stewart, Lisa Yung
- **Others involved:** Eugene Charniak, Dustin Hillard, Mark Johnson, Jeremy Kahn, Mari Ostendorf, Andreas Stolcke, Liz Shriberg, Wen Wang, Ann Bies and the LDC Treebanking Team
- **Our sponsors and everyone at CLSP!**

Contributions

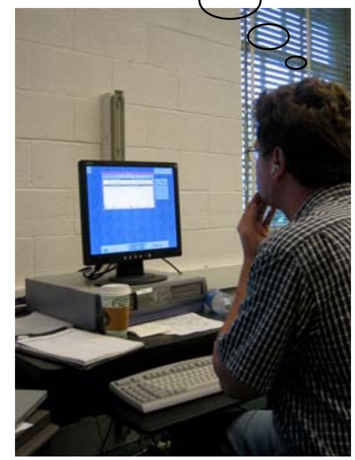
- Dev1, Dev2, and Eval treebanks consistent with MDE annotations
- Sparseval tool to evaluate speech parse accuracy; alignment tool
- Tools and scripts for cleaning, annotating, and transforming trees
- Feature extraction tools
- Reranking framework for SU
- Solid results and an excellent basis for future research!!

Cleaning tool!

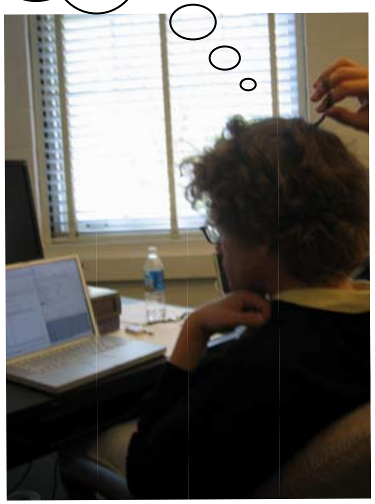


Minipar convert!

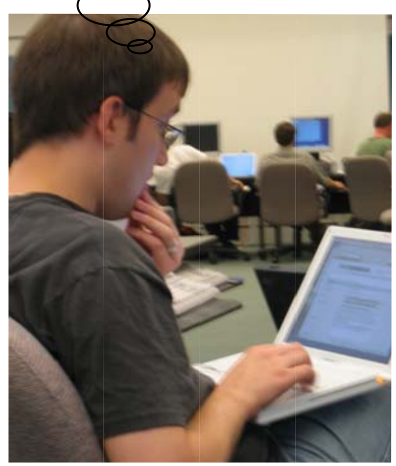
Sparseval!



Editannot!



Alignment and Dependency extraction!



Segmentation!