

ICSI-SRI-UW RT03F MDE System and Research

Yang Liu, Chuck Wooters, Barbara Peskin

ICSI

Elizabeth Shriberg, Andreas Stolcke

SRI & ICSI

Dustin Hillard, Mari Ostendorf

UW



Talk Outline

Structural MDE (Liz, Yang, Dustin)

- Summary of submitted systems, with updated results
- System descriptions
- Post-evaluation fixes and enhancements
- Research
 - ◻ Sampling techniques for prosodic classifiers
 - ◻ Effect of speaker segmentation on BN SU detection
 - ◻ Effect of better STT on MDE
 - ◻ Effect of using TB3 data
 - ◻ Confusion networks for MDE decoding (UW)
 - ◻ Other things we tried
- Summary

Diarization (Chuck)

- System description
- Analysis of results
- Some CTS experiments
- Future Work



Structural MDE: Overview

- ▣ We submitted results for all tasks and conditions
- ▣ Official results due just as scoring tools stabilized
- ▣ Since then, we've been able to
 - Fix problems
 - Interact with other sites (very fruitful)
 - Do more research
- ▣ Thus we have many improved results
- ▣ Given all the tasks we participated in, chose to focus on one for post-eval effort. Chose SU task (events more frequent, more participants)



Tasks and Results (Eval Post-eval)

	NIST-score BBN-score					BBN-score	
	SUBD	<i>SUBD Contrast</i>	EWD	FWD	IPD	SASTT	03RT
BN	49.04 49.04	48.72 48.72	44.20 44.20	7.91 7.91	15.22 15.65	15.65	19.07
	48.83 48.83					9.63	13.21
spch	59.28 60.13	57.25 58.10	96.69 96.13	46.76 50.36	65.76 64.76	19.46	29.19
	58.00 58.85					13.47	23.28
CTS	30.13 30.13		59.03 58.99	18.53 18.24	27.51 27.48	0	10.07
	28.17 28.17		58.79 58.76	17.89 17.71	26.90 26.88		9.73
spch	45.53 46.67		87.86 87.48	45.54 46.86	63.49 63.18	10.81	33.23
	43.58 44.55		87.44 87.01	45.58 46.93	63.46 63.24		33.09

Note: These results don't include the recent UW confusion network results [discussed below]



System Description: Statistical Modeling

Language models:

- 4-gram LM over words and metadata events (KN smoothing)
- Class-based LM (classes induced from bigram statistics, WB smoothing)
- POS LM (POS tags from TnT tagger, WB smoothing)
- Repetition string LM

Prosody model: decision tree classifier

- Has available about 100 features (duration, F0, energy, pause)
- Yields posterior probability of events at each word or inter-word boundary, given the prosodic features
- New work to deal with skewed class distribution (sampling of training data; bagging and ensemble of trees)



System Description: Model Combination

Words and hidden event labels define HMM

- Transition probabilities given by LM
- Observation likelihoods derived from prosodic classifier output

HMM parameter training fully supervised

- Using MDE-labeled training data

Test data decoded using forward-backward algorithm

- Finds event with highest posterior probability at each word boundary



System Description: Test Data Processing

A Acoustic segmentation (chunking)

- Ref: segment the speech data using the provided time marks (break at long pauses)
- Spch: same segments as used as input to STT system

F Forced alignment

- Produces word and phone time marks

S Speaker labeling

- BN: automatic clustering as used in STT system (different from speaker diarization results)
- CTS: based on channel

C Compute prosodic features



CTS SU System

Binary classification: SU vs. not SU

LMs

- Hidden-event word-based 4-gram LM trained from LDC data
- Hidden-event word-based 4-gram LM trained from Meteer data
- Hidden-event automatically-induced class-based 4-gram LMs trained from LDC data
- Part-Of-Speech based 5-gram LM trained from LDC data

Combination

- Interpolate the word-based LMs and the class-based LM
- Combine posterior probabilities from prosody model with resulting LM using an HMM
- For ref condition only: Run POS-LM separately. Then interpolate with the resulting model above. (No win for spch condition)



BN SU System

Two-way classification: SU vs. not SU

LMs

- Hidden-event word-based LM trained from the new LDC data
- A large LM derived from STT LM (by replacing <s> with SU)
- Hidden-event automatically-induced class-based LM trained from new LDC data (only used for ref condition, found no win for spch condition)

Combination

- All LMs are interpolated
- LMs and prosody model are combined using the HMM approach



Edit Word System

Used prosody model only for CTS (not enough time, data for BN)

For CTS only, IP detection: two-way classification, prosody model combined with the hidden-event LM trained from the LDC data

For both BN and CTS:

- All word fragments generate IP candidates
- Repetition detection: look for matched strings (with possible intervening fillers)
- Edit word detection: working backward from IPs, look for word(s) that match the word following the IP (allowing fragments or contractions). Captures “retracing”
- If no ‘valid’ edit region is found this way, see if IP coincides with SU (but not Turn) boundary. If so, hypothesize edit (restart)
- If not a restart and there are no matched words, delete the IP hypothesis



Filler Word System

- Two-way classification for filled pause: FP vs. not FP
- Two-way classification for end of discourse marker: DM vs. not DM
- Combine prosody model and hidden-event LM
- When end of DM is hypothesized, work backward to check whether word sequence is on the DM list (DMs can be > 1 word)
- For ref condition, also use FP subtype information provided
(Note: this should be perfect but was not. Also not clear should be provided)

IP System

- Final IPs are generated from the edit and filler results using NIST's ctm+mdtm-to-rttm script



Reconciling Conflicting MD Events

Our by-task submissions were based on our RT submission

For RT, conflicting events need to be resolved:

- At present, conflict affects only the 2 boundary types (SU and IP):
 - repetition or SU?: that's great * that's great
 - restart or inc.SU?: so you're * are you from texas
- In general could affect any types (including word-based events)

Currently we resolve conflicts in RT using *ad hoc* approach:

- Set threshold on SU prob, since we're more confident in that
- For eval submission, SU prob had to beat threshold of .85 to win

Correct approach would be to jointly optimize for all subsystems
(expensive; future work)



Post-Evaluation System Fixes: Word Chunking for CTS SPCH

- Acoustic segmentation of the word stream into chunks affects the way features are extracted
- In ref condition and for training data, chunks based on pauses
- For CTS spch, we had used chunks from STT, causing a mismatch
- We redid chunking for CTS spch condition to match training
- Improvement: roughly 1% absolute

Note: collaboration with Cambridge alerted us to this problem □



Fixes: Consistent Acoustic Models for Forced Alignments

- Our STT system used more recent acoustic models than did our MDE alignments
- Older models were used originally to align both training data and ref output
- We realized this before the eval, and updated alignments for ref output, but not for training data (lack of time)
- Thus prosody model had a known mismatch
- Using consistent acoustic models probably resulted in a small post-eval improvement (haven't run clean contrast yet)



Fixes: Text Normalization for CTS SU

Fixed problems with mapping of spellings of certain words (e.g., uhhuh, filled pauses, and other items) to match SRI recognizer

- Affected spch condition -- words output by STT system did not match those in the MDE LM
- Also affected ref condition – since “reject” models are used for words with missing pros; prosodic features for such cases were not correct

This resulted in roughly a 1% improvement for CTS SU detection

But: We haven't fixed all text normalization yet

- Haven't fixed BN yet (since we interpolate MDE SU LM with STT LM, this means estimates for words spelled differently not combined correctly)
- Haven't rerun other CTS tasks yet



Prosody Model: Coping with Skewed Data (1)

Skewed class distributions (= imbalanced priors) present common problem for MDE

Investigated different sampling approaches:

- Random downsampling: randomly downsample the majority class to get balanced training set
- Oversampling: replicate minority class samples
- Ensemble downsampling: split the majority class into N subsets, each of which is combined with the minority class to train a decision tree; combine probs from resulting trees
- SMOTE (synthetic minority over-sampling): generate synthetic samples in the neighborhood of the existing minority class
examples [derived from Nitesh Chawla, "Synthetic Minority Over-sampling Technique", Journal of AI Research, 2002]
- Original training set: no sampling is used. Need to adjust priors when combined with LMs



Coping with Skewed Data (2)

Bagging:

- Bagging combines classifiers trained from different samples (with replacement) given a training set
- Different classifiers make different errors. Combining them yields superior performance to that of a single classifier
- Bagging is also computationally efficient and training can be parallelized
- We used bagging on the downsampled training set (for efficiency reasons)

Ensemble bagging:

- Split majority class, combine with the minority class to make multiple balanced training sets
- For each subset, apply bagging
- Parallel training



Coping with Skewed Data (3)

- Experimented with SU task for CTS
- Split the LDC-1st CTS training set into training and testing subsets
- Scored using legacy SRI SU boundary error scoring

	Prosody Alone	Prosody+LM
Chance	13.00	-
Downsampling	8.48	4.14
Oversampling	10.67	4.39
Ensemble downsampling	7.61	4.18
SMOTE	8.05	4.31
Original	7.32	4.08
Bagging (on downsampled)	7.10	3.98
Ensemble bagging	6.93	3.89



Prosody Model: Conclusions (1)

Sampling:

- Among sampling approaches, downsampling gives close to the best results when combined with LM and allows faster training because training set size is reduced
- Using the original training set achieves the best performance in this experiment --- BUT, it is a small training set! Training a decision tree from a large imbalanced training set is very computationally demanding
- Using all the training set in an ensemble way can reduce the variance due to the random downsampling. Training can run in parallel
- Oversampling (with replication or SMOTE) yields no gain and increases computational cost



Prosody Model: Conclusions (2)

Bagging

- Bagging improves the generalization of decision tree classifier; it mitigates the overfitting of a single classifier and reduces variance by averaging the diff trees
- Ensemble bagging (bagging disjoint subsets of data) makes full use of all the training data. It is computationally efficient and scalable to large training set

Next steps

- So far ensemble of classifiers is done based on different training set and different sampling. We plan to build ensemble of trees using different prosodic feature sets as well
- We plan to investigate a two-pass process: first allow more false positives, then use other knowledge sources to narrow final decision



Speaker Diarization & SU Recognition

- Speaker turn change is a strong indicator for SU in prosody model
- Also affects the input to LM (we consolidate all segments within the same speaker into a single chunk for the LM)
- BN ref condition: (using our eval submission system)

Speaker segmentation	su-eval
Automatic clustering (from STT)	49.04
Speaker diarization results (*)	48.72
Reference speaker info	45.52

(*) the diarization results are based on a preliminary run (not the sastt submission)

⇒ Correct speaker info is important for SU task



STT Accuracy and MDE

▣ Tested effect of different STT output on CTS spch SU task

▣ Many thanks to CUED for providing their STT output!

▣ WER is not the only factor:

- Sentence initial or final words are especially important
- Quality of STT speech detection could be crucial

STT system	su-eval	rteval (subd)	rt1
SRI	43.58	44.55	24.89
CUED	39.89	41.58	22.83



Mapped TB3 (Meter) Data

ICI “crudely” mapped TB3-annotated data to V5-like annotation

Provided detailed information on differences between schemes

Note: These differences reflect philosophical differences and/or mismatch between ISIP transcripts and original TI transcripts

TB3 set larger than V5, but to benefit from it requires:

- Methods for combining it “as is” with V5
- Using it for tasks in which mismatch is minimized
- Methods to make TB3 look more like V5 (research area)



Effect of Adding TB3 to Training

So far our team has only tried using TB3 “as is”

Results to date are MIXED; depend on task and approach

UW found that adding TB3 data to V5 data, unweighted:

- *hurt* SU performance and showed *mixed* results for IP detection
- *helped* edit and filler detection (using TBL)

ICSI tried only for SU task; combined via weighted interpolation:

- found adding TB3 *helps*, for example on CTS ref:

	V5	V5+TB3
Word-LM: <i>+prosody with bagging</i>	30.97	29.72

More research needed on how to use this data

- Seems weighting is necessary
- Need to improve mapping beyond the limited rule-based method we employed. Machine learning could help



Contributions from Individual Components to Post-Eval Result for CTS SU Ref Task

Component	su-eval
LDC-LM alone	38.08
add prosody (downsampled)	33.24
employ bagging for prosody	30.97
add Meteer data LM	29.72
add class LM	28.97
add POS LM	28.64
employ ensemble for bagged prosody	28.19
add prosody model from Meteer data	28.17



Confusion Networks for MDE

Task:

SU detection for BN and CTS

Long term goal:

More integrated STT and MDE

Hypothesis:

Looking at multiple STT hyps leads to improvements in MDE detection and hopefully STT



Example: Multiple Hyps Are Helpful

REF ANY easier for the president ** OR *** the united states .

1BEST: AN easier for the president OF WAR []. the united states .

2BEST: any easier for the president OF WAR []. the united states .

3BEST: AN easier for the president OR the united states .

4BEST: any easier for the president OR the united states .

5BEST: AN easier for the president AT WAR []. the united states .



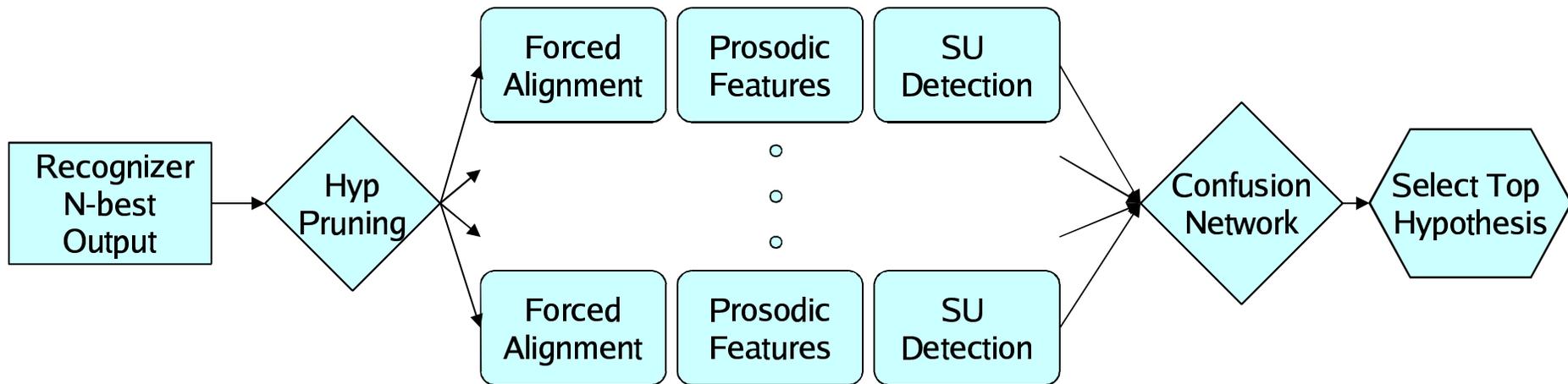
Should be no SU here.

Idea:

If the detected SU in hyp 1 has low probability, then the combined SU-STT score could raise the ranking of the correct hyp.



SU Confusion Network System



Hypothesis Pruning

Processing for each hyp requires:

- Forced alignment run
- Prosodic feature extraction
- SU detection

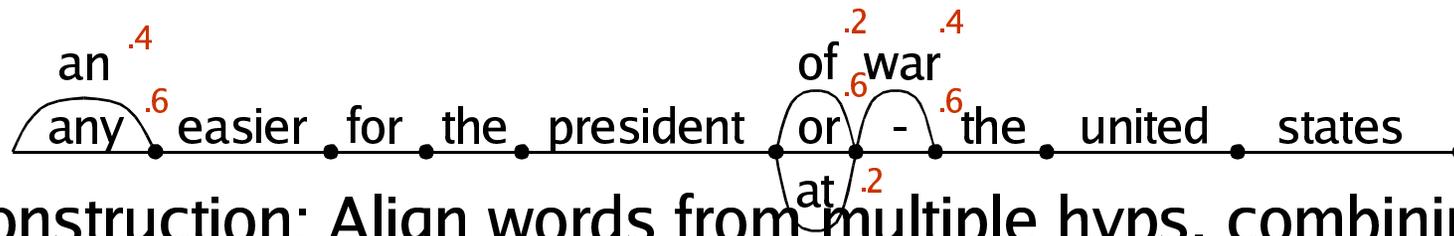
Reduce computation by selecting high confidence hyps

- Take just the top 90% of N-best hyps (by posterior mass)
- Stop at 1000 hyps if 90% mass not reached



ASR Confusion Networks

Confusion networks for ASR

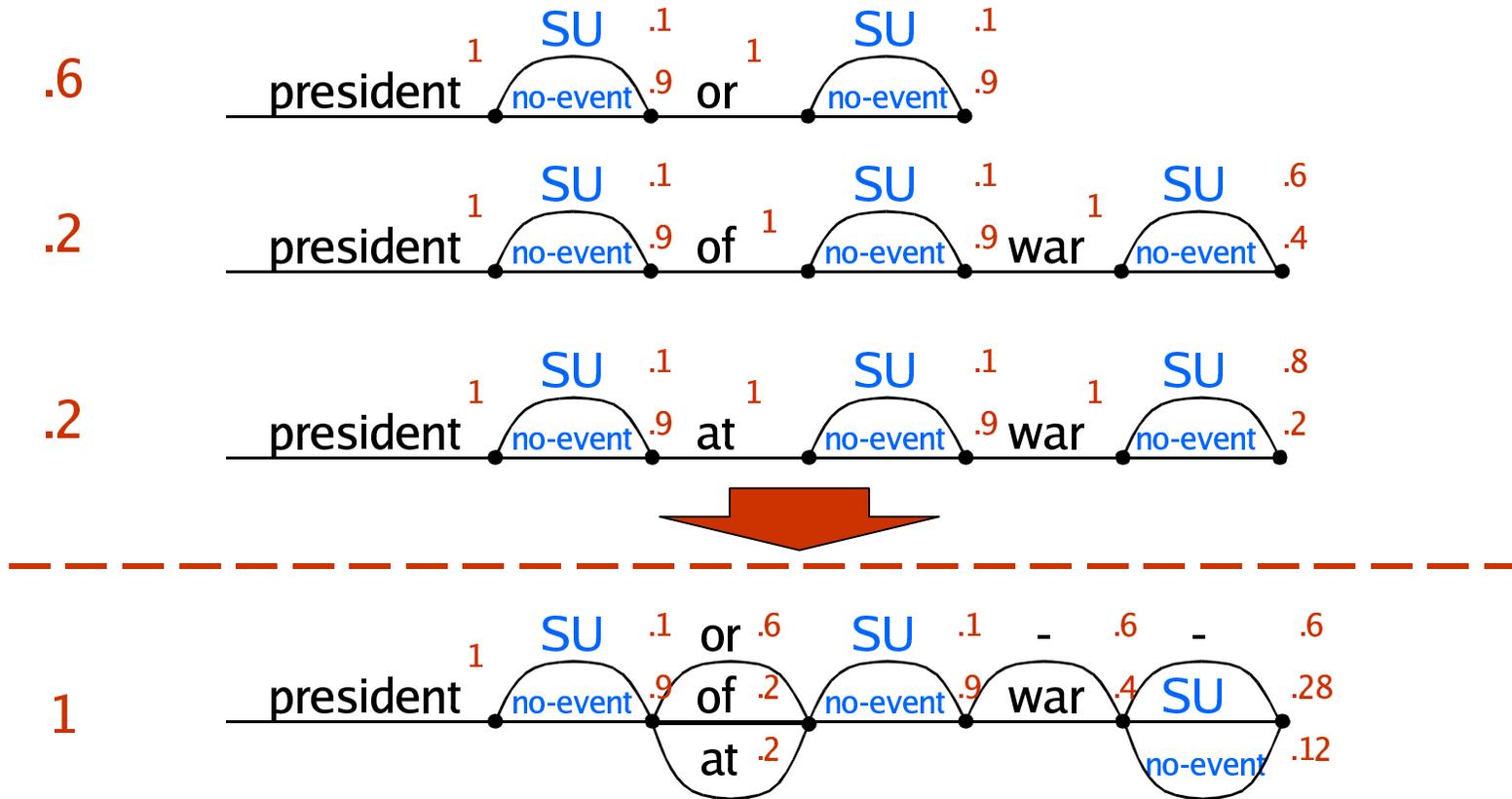


Construction: Align words from multiple hyps, combining weights for co-occurring words

Decoding: Select best word at each slot

SU Confusion Networks

Include SU events in word confusion networks



SU Task Results: Single Hyp vs. Confusion Nets

su-eval / rteval

	BN Dev	BN Eval	CTS Dev	CTS Eval
Pruned Rover Hypothesis	55.79% / 56.52%	57.78% / 58.85%	44.14% / 48.12%	44.95% / 48.53%
Confusion Networks	54.45% / 54.81%	57.68% / 58.10%	43.06% / 46.95%	44.42% / 47.94%

All results based on same set of pruned hyps

Pruning hurts in most cases; gain from confusion nets compensates for BN but not CTS



Future Work with Multiple STT Hyps

- Optimize N-best prosodic feature computation
- Alternative confusion network configurations that would also impact WER
- Move from N-best to lattice representations of STT output
- Leverage prior SRI work on discriminative score combination
- Extend confusion nets to other structural MDE



Structural MDE: Things We Also Tried (1)

Factored language modeling (code: Bilmes & Kirchhoff, UW)

- Way to model interaction between words and prosody
- Discretized prosodic features are attached to words
- So far: better than LM alone, but not than LM+standard prosody model

Breath modeling for BN

- ‘Inspiration’: breaths in BU radio news at major phrase boundaries
- Trained breath models using Hub4 training data (breaths marked)
- Ran forced alignment with optional breaths between words for BN
- Used posterior prob of breath in prosodic decision tree
- So far no gain. Feature is chosen, but doesn’t help overall accuracy
- Unclear how accurate both the breath marks and models are



Things We Tried (2)

Speaking-style-dependent prosodic modeling

- Idea: same MD event can be rendered different ways (words, prosody)
- Just started to investigate for prosodic modeling for CTS
- Used simple automatic metrics (e.g. rate of speech) as features in tree
- Trees show such features *do* condition other features, but so far no win
- There may be better approaches

Compact feature sets for prosodic modeling

- Idea: smaller sets of features can perform better than large sets
- Used modified feature selection algorithm; found smaller set that performs better on both dev and eval data
- Small win even after combining with LM
- However: limiting the feature list decreases the gain from ensemble bagging; smaller feature list allows less variability in feature usage across trees



Things We Tried (3)

▣ **TBL for SU and edit detection** (inspired by Schwarm & Ostendorf, UW)

- Learn rules based on words and POS to iteratively relabel data
- Looks promising, but so far no win over standard statistical classifier

▣ **Use of other LM training data for CTS SU modeling**

- Fisher (quick transcriptions): inferred SU boundaries from punctuation
- Web data: inferred SU boundaries using max-ent sentence tagger
- Interpolated resulting LMs with standard SU LMs
- No improvements (yet)
- Need better mapping for different SU definitions

▣ **Different confusion net topologies**

- Link words and MDE in a single arc to more tightly couple decisions
- In pilot experiments, leads to gain in BN but hurts in CTS



Future Work

BN

- Filler detection (especially discourse markers) may use the models from CTS
- Further explore impact of speaker segmentation on SU task
- Use syntactic structure to find possible SUs

CTS

- Use more sophisticated models for edit word detection
- Make better use of prosody for filler task detection
- Investigate how to make better use of TB3 data

Jointly optimize component classifiers

Continue exploring MDE + STT interaction via confusion networks

Community action item: scoring fixes

- Our diagnostics uncovered a number of small problems with scorers (best deferred to tools discussion)



Structural MDE: Summary

Submitted systems for all tasks

Fixed some problems post-eval; made significant improvements

Modeling:

- Started with previous “hidden event” modeling approach
- Added new types of LMs and new ways of combining them (with each other and with prosody model)
- Significantly improved prosody model using sampling and ensemble approaches to deal with inherent skew in class sizes
- Introduced confusion network combination of SU and sentence hypotheses in a move towards more integrated STT & MDE

Ran diagnostic experiments to explore effects of:

- Diarization on SU detection
- STT accuracy on MDE
- Adding mismatched training data (TB3)

Began to explore a number of other interesting ideas

