Automatic Assessment of Non-Native Prosody

Annotation, Modelling and Evaluation

Florian Hönig
June 7th, 2012
Computer Science Dept. 5 (Pattern Recognition Lab)

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Non-Native Prosody

- A main source for low intelligibility of learner

Non-Native Prosody

- A main source for low intelligibility of learner

[*We're planning to travel to Egypt for a week or so.*]

Non-Native Prosody

- A main source for low intelligibility of learner


[*We're planning to travel to Egypt for a week or so.*]

Non-Native Prosody

- A main source for low intelligibility of learner
  - [*We're planning to travel to Egypt for a week or so.*]
  - [*Can I take a message?*]

Non-Native Prosody

- A main source for low intelligibility of learner
  - [*We're planning to travel to Egypt for a week or so.*]
  - [*Can I take a message?*]
  -

Non-Native Prosody

- A main source for low intelligibility of learner
  - [*We're planning to travel to Egypt for a week or so.*]
  - [*Can I take a message?*]
  - [*I'm afraid I'm playing tennis on saturday.*]

Non-Native Prosody

- A main source for low intelligibility of learner
  - [*We're planning to travel to Egypt for a week or so.*]
  - [*Can I take a message?*]
  - [*I'm afraid I'm playing tennis on saturday.*]
- Word accent position, syntactic-prosodic boundaries, and rhythm: help listener to process segmental, syntactic, and semantic content
- Most disruptive: Strange rhythm

→ Prosody really relevant for mutual understanding, and thus for CAPT

Outline

- Machine learning and evaluation
- Annotation
- Automatic Assessment: Target & Approach
- Data
- Features
- Modelling
- Experiments & Results
- Conclusion


Ignorance
is Bliss!

Machine Learning

- The pattern recognition approach – collecting annotated data, extracting suitable features, and applying machine learning for classification/regression – is indispensable for CAPT
- Fundamental postulate of pattern recognition:

  For collecting information on a task, a **representative** *corpus of training samples (instances) is available.*

Machine Learning

- The pattern recognition approach – collecting annotated data, extracting suitable features, and applying machine learning for classification/regression – is indispensable for CAPT
- Fundamental postulate of pattern recognition:

  For collecting information on a task, a **representative** *corpus of training samples (instances) is available.*

- In CAPT, data is usually **far** from being representative

Machine Learning

- The pattern recognition approach – collecting annotated data, extracting suitable features, and applying machine learning for classification/regression – is indispensable for CAPT
- Fundamental postulate of pattern recognition:

  For collecting information on a task, a **representative** *corpus of training samples (instances) is available.*

- In CAPT, data is usually **far** from being representative

→ Usually, **overfitting** will occur, i. e. performance on new data is (possibly much) lower than on collected data

Overfitting

- Reduce
  - Simpler models
  - Better machine learning algorithms
- Anticipate: evaluate performance on **unseen data**
  - Strict division of instances into train and test set
  - If data is sparse, use cross-validation

Overfitting (2)

- Popular procedure
  - Set up scheme to perform cross-validation
  - Optimize system for best performance in cross-validation

Overfitting (2)

- Popular procedure
  - Set up scheme to perform cross-validation
  - Optimize system for best performance in cross-validation

  → System is effectively tuned on known data; overfitting may still occur!

Overfitting (2)

- Popular procedure
  - Set up scheme to perform cross-validation
  - Optimize system for best performance in cross-validation

  → System is effectively tuned on known data; overfitting may still occur!
- Generally, all decisions taken **using the collected data** are possibly prone to overfitting:
  Modelling approach, classifier, classifier parameters, classifier meta-parameters, implemented features, selected features, . . .

Overfitting (2)

- Popular procedure
  - Set up scheme to perform cross-validation
  - Optimize system for best performance in cross-validation

  → System is effectively tuned on known data; overfitting may still occur!
- Generally, all decisions taken **using the collected data** are possibly prone to overfitting:
  Modelling approach, classifier, classifier parameters, classifier meta-parameters, implemented features, selected features, . . .

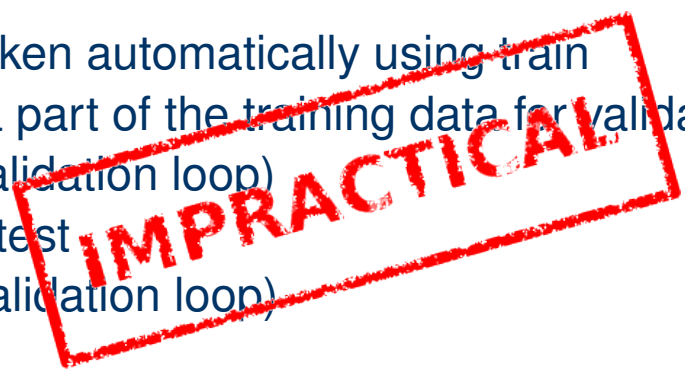
→ Ideally:
  - Set up **all** decisions to be taken automatically using train
  - Where necessary, reserve a part of the training data for validation (e.g. using an inner cross-validation loop)
  - Evaluatate performance on test (e.g. using an outer cross-validation loop)

Overfitting (2)

- Popular procedure
  - Set up scheme to perform cross-validation
  - Optimize system for best performance in cross-validation

  → System is effectively tuned on known data; overfitting may still occur!
- Generally, all decisions taken **using the collected data** are possibly prone to overfitting:
  Modelling approach, classifier, classifier parameters, classifier meta-parameters, implemented features, selected features, . . .

→ Ideally:
  - Set up **all** decisions to be taken automatically using train
  - Where necessary, reserve a part of the training data for validation (e.g. using an inner cross-validation loop)
  - Evaluatate performance on test (e.g. using an outer cross-validation loop)



Overfitting (3)

- Compromise
  - Properly evaluate the really bad guys (or avoid them)
  - Optimize the remaining parameters on **test**,
    but be **verbose** about it

Overfitting (3)

- Compromise
  - Properly evaluate the really bad guys (or avoid them)
  - Optimize the remaining parameters on **test**,
    but be **verbose** about it

- What is a bad guy? No general answers, but consider . . .
  - . . . # instances / # data points vs. # parameters / # models
  - . . . distrusting results like 99% recognition rate
  - . . . consulting a fellow researcher
  - . . . experimentation

Overfitting Revisited

- Imagine:

  Cross-validated everything correctly, got 90% recognition rate, but . . .

Overfitting Revisited

- Imagine:

  Cross-validated everything correctly, got 90% recognition rate, but . . .

  . . . customer complains that your classifier is faulty!

Overfitting Revisited

- Imagine:

  Cross-validated everything correctly, got 90% recognition rate, but . . .

  . . . customer complains that your classifier is faulty!

  . . . new data collection from application proves that!!

Overfitting Revisited

- Imagine:

  Cross-validated everything correctly, got 90% recognition rate, but . . .

  . . . customer complains that your classifier is faulty!

  . . . new data collection from application proves that!!

- Just testing on unseen **instances** is not enough!
  $\rightarrow$ design test set to reflect realistic **application** data
- Examples:
  - Unseen speakers
  - Unseen recording conditions
  - Unseen texts

Cross-Validation Hell

- Multiple independent conditions: cross-validation gets wasteful . . .
- Example: Speaker- **and** text-independent evaluation
  (*N*-fold crossvalidation, *N* speakers, *N* sentences):

```
for i = 1 ... N    // test speaker
  for j = 1 ... N  // test sentence
    train := all except speaker i AND except sentence j
    test := only speaker i AND sentence j
    // train and eval:
    ...
  endfor
endfor
```

Cross-Validation Hell

- Multiple independent conditions: cross-validation gets wasteful . . .
- Example: Speaker- **and** text-independent evaluation
  (*N*-fold crossvalidation, *N* speakers, *N* sentences):

```
for i = 1 ... N    // test speaker
  for j = 1 ... N  // test sentence
    train := all except speaker i AND except sentence j
    test := only speaker i AND sentence j
    // train and eval:
    ...
  endfor
endfor
```

- K conditions $\rightarrow$ size of train $\left(\frac{N-1}{N}\right)^K$; # iterations: N^K
  In example: Train with $\geq 50\% \rightarrow N \geq 4$ ($4^2 = 16$ iterations)

Cross-Validation Hell

- Multiple independent conditions: cross-validation gets wasteful . . .
- Example: Speaker- **and** text-independent evaluation
  (N-fold crossvalidation, N speakers, N sentences):

```
for i = 1 ... N    // test speaker
  for j = 1 ... N  // test sentence
    train := all except speaker i AND except sentence j
    test := only speaker i AND sentence j
    // train and eval:
    ...
  endfor
endfor
```

- K conditions $\rightarrow$ size of train $\left(\frac{N-1}{N}\right)^K$; # iterations: N^K
  In example: Train with $\geq 50\% \rightarrow N \geq 4$ ($4^2 = 16$ iterations)
- With validation set: Size of train $\left(\frac{N-1}{N}\right)^{2K}$, # iterations: N^{2K}
  In example: Train with $\geq 50\% \rightarrow N \geq 7$ ($7^4 = 2401$ iterations)

Challenges

Challenges

- Organizer defines task and evaluation scheme, and provides data
  $\rightarrow$ results are, for once, **comparable**!
- Examples:
  The INTERSPEECH 2009 Emotion Challenge
  The INTERSPEECH 2010 Paralinguistic Challenge
  The INTERSPEECH 2011 Speaker State Challenge
  The INTERSPEECH 2012 Speaker Trait Challenge
  SLaTE 2013 Challenge(?)

Challenges

- Organizer defines task and evaluation scheme, and provides data
  $\rightarrow$ results are, for once, **comparable**!
- Examples:
  The INTERSPEECH 2009 Emotion Challenge
  The INTERSPEECH 2010 Paralinguistic Challenge
  The INTERSPEECH 2011 Speaker State Challenge
  The INTERSPEECH 2012 Speaker Trait Challenge
  SLaTE 2013 Challenge(?)
- Advantages for participants
  - Dataset is provided

Challenges

- Organizer defines task and evaluation scheme, and provides data
  $\rightarrow$ results are, for once, **comparable**!
- Examples:
  The INTERSPEECH 2009 Emotion Challenge
  The INTERSPEECH 2010 Paralinguistic Challenge
  The INTERSPEECH 2011 Speaker State Challenge
  The INTERSPEECH 2012 Speaker Trait Challenge
  SLaTE 2013 Challenge(?)
- Advantages for participants
  - Dataset is provided
  - Objective proof for performance of your algorithm

Challenges

- Organizer defines task and evaluation scheme, and provides data
  $\rightarrow$ results are, for once, **comparable**!
- Examples:
  The INTERSPEECH 2009 Emotion Challenge
  The INTERSPEECH 2010 Paralinguistic Challenge
  The INTERSPEECH 2011 Speaker State Challenge
  The INTERSPEECH 2012 Speaker Trait Challenge
  SLaTE 2013 Challenge(?)
- Advantages for participants
  - Dataset is provided
  - Objective proof for performance of your algorithm
- Advantages for organizer
  - Customers: Get the objectively best method

Challenges

- Organizer defines task and evaluation scheme, and provides data
  $\rightarrow$ results are, for once, **comparable**!
- Examples:
  The INTERSPEECH 2009 Emotion Challenge
  The INTERSPEECH 2010 Paralinguistic Challenge
  The INTERSPEECH 2011 Speaker State Challenge
  The INTERSPEECH 2012 Speaker Trait Challenge
  SLaTE 2013 Challenge(?)
- Advantages for participants
  - Dataset is provided
  - Objective proof for performance of your algorithm
- Advantages for organizer
  - Customers: Get the objectively best method
  - Citations $\uparrow\uparrow$

Challenges: How to Organize Evaluation?

- Method 1:
  - Provide whole dataset,
  - Participant sends code,
  - Organizer runs cross-validation

Challenges: How to Organize Evaluation?

- ~~Method 1:~~
  - ~~Provide whole dataset~~
  - ~~Participant sends code~~
  - ~~Organizer runs cross-validation~~
- Divide data into **train, develop** and **test** set

Challenges: How to Organize Evaluation?

- ~~Method 1:~~
  - ~~Provide whole dataset~~
  - ~~Participant sends code~~
  - ~~Organizer runs cross-validation~~
- Divide data into **train, develop** and **test** set
- Method 2:
  - Provide only train and develop (hold back test)
  - Participant sends code,
  - Organizer runs training & evaluation

Challenges: How to Organize Evaluation?

- ~~Method 1:~~
  - ~~Provide whole dataset~~
  - ~~Participant sends code~~
  - ~~Organizer runs cross-validation~~
- Divide data into **train, develop** and **test** set
- Method 2:
  - Provide only train and develop (hold back test)
  - Participant sends code,
  - Organizer runs training & evaluation

  $--$ expensive $+$ totally save

Challenges: How to Organize Evaluation?

- ~~Method 1:~~
  - ~~Provide whole dataset~~
  - ~~Participant sends code~~
  - ~~Organizer runs cross-validation~~
- Divide data into **train, develop** and **test** set
- Method 2:
  - Provide only train and develop (hold back test)
  - Participant sends code,
  - Organizer runs training & evaluation

  $--$ expensive $+$ totally save
- Method 3:
  - provide whole dataset (hold back labels for test)
  - participant sends only results for test

Challenges: How to Organize Evaluation?

- ~~Method 1:~~
  - ~~Provide whole dataset~~
  - ~~Participant sends code~~
  - ~~Organizer runs cross-validation~~
- Divide data into **train, develop** and **test** set
- Method 2:
  - Provide only train and develop (hold back test)
  - Participant sends code,
  - Organizer runs training & evaluation

  $--$ expensive $+$ totally save
- Method 3:
  - provide whole dataset (hold back labels for test)
  - participant sends only results for test

  $++$ practical $-$ theoretical chance for cheating

Challenge Evaluation Hell

- Need to partition data into train, develop and test
  — disjunct w. r. t. conditions?
- Example: Speakers S1, S2, S3; sentences T1, T2, T3:

speaker	sentence	set
S1	T1	
S1	T2	
S1	T3	
S2	T1	
S2	T2	
S2	T3	
S3	T1	
S3	T2	
S3	T3	

- Fraction of used data: $\left(\frac{1}{3}\right)^{K-1}$

Challenge Evaluation Hell

- Need to partition data into train, develop and test
  — disjunct w. r. t. conditions?
- Example: Speakers S1, S2, S3; sentences T1, T2, T3:

speaker	sentence	set
S1	T1	train
S1	T2	
S1	T3	
S2	T1	
S2	T2	
S2	T3	
S3	T1	
S3	T2	
S3	T3	

- Fraction of used data: $\left(\frac{1}{3}\right)^{K-1}$

Challenge Evaluation Hell

- Need to partition data into train, develop and test
  — disjunct w. r. t. conditions?
- Example: Speakers S1, S2, S3; sentences T1, T2, T3:

speaker	sentence	set
S1	T1	train
S1	T2	
S1	T3	
S2	T1	
S2	T2	develop
S2	T3	
S3	T1	
S3	T2	
S3	T3	

- Fraction of used data: $\left(\frac{1}{3}\right)^{K-1}$

Challenge Evaluation Hell

- Need to partition data into train, develop and test
  — disjunct w. r. t. conditions?
- Example: Speakers S1, S2, S3; sentences T1, T2, T3:

speaker	sentence	set
S1	T1	train
S1	T2	
S1	T3	
S2	T1	
S2	T2	develop
S2	T3	
S3	T1	
S3	T2	
S3	T3	test

- Fraction of used data: $\left(\frac{1}{3}\right)^{K-1}$

Avoiding Hell

- Anticipate conditions **when collecting data**

Avoiding Hell

- Anticipate conditions **when collecting data**

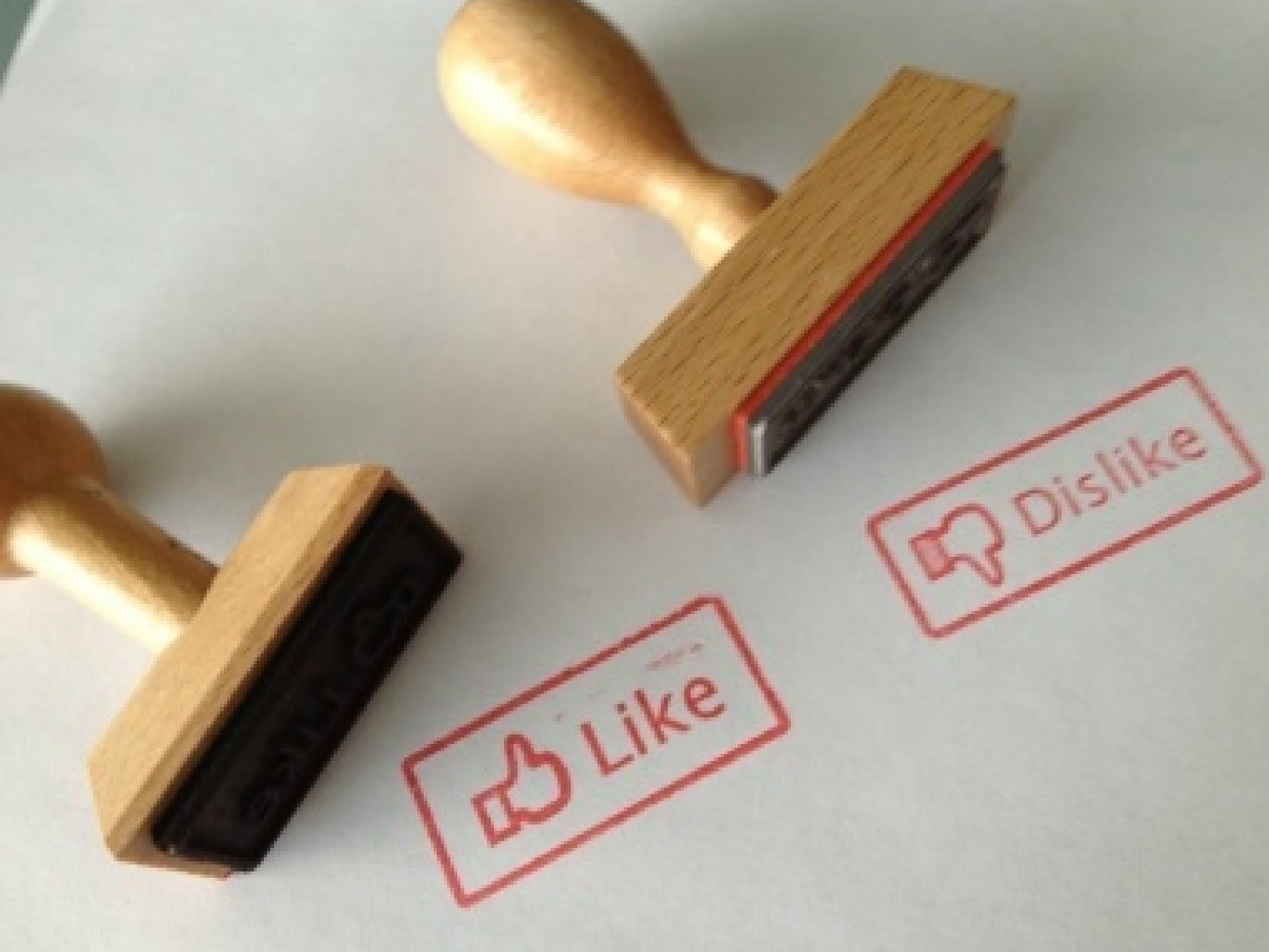
speaker	sentence	...	set
S1	T1		
S1	T1		
S1	T1		
S2	T2		
S2	T2		
S2	T2		
S3	T3		
S3	T3		
S3	T3		

Avoiding Hell

- Anticipate conditions **when collecting data**

speaker	sentence	...	set
S1	T1		train
S1	T1		train
S1	T1		train
S2	T2		develop
S2	T2		develop
S2	T2		develop
S3	T3		test
S3	T3		test
S3	T3		test

- Better: Use 4 partitions to provide validation set for participants


Like
Dislike

Multiple Labellers

- For evaluating CAPT approaches, we need annotated data
  - Individual labellers will make errors
  - Combining multiple labellers will help
  - How many?

Multiple Labellers

- For evaluating CAPT approaches, we need annotated data
  - Individual labellers will make errors
  - Combining multiple labellers will help
  - How many?
- Here: Aiming at continuous labels
- A^N: Reference created by averaging over the annotations of N labellers
- Ground truth: Combining infinitely many labellers: A^∞
- **Quality** of A^N: Correlation to ground truth

Multiple Labellers

- For evaluating CAPT approaches, we need annotated data
  - Individual labellers will make errors
  - Combining multiple labellers will help
  - How many?
- Here: Aiming at continuous labels
- A^N: Reference created by averaging over the annotations of N labellers
- Ground truth: Combining infinitely many labellers: A^∞
- **Quality** of A^N: Correlation to ground truth

$$\mathrm{Corr}(A^N, A^\infty) = \sqrt{c/(\frac{1}{N} + \frac{N-1}{N}c)}$$

- c: Average pair-wise correlation

Multiple Labellers - Example

- Average pairwise correlation: $c = 0.3$
- Average quality of a single labeller: $\sqrt{c} = 0.55$
- Quality of combined annotation from 10 labellers: 0.90

Types of Labellers

- Experts
  - Better, but more expensive
  - May be biased
- Naïves
  - Less consistent, but cheaper
  - Task has to be well-defined
  - Recent trend towards using crowdsourcing

Types of Labellers

- Experts
  - Better, but more expensive
  - May be biased
- Naïves
  - Less consistent, but cheaper
  - Task has to be well-defined
  - Recent trend towards using crowdsourcing

- Questions:
  - When comparing different types of labellers, how large are inter- and intragroup effects?
  - Quality of combined annotations of Group $\mathbb{A}$ with respect to group $\mathbb{B}$?

Types of Labellers (2)

- Average pair-wise correlation within group $\mathbb{A}$: *c*
- Average pair-wise correlation within group $\mathbb{B}$: *d*
- Average pair-wise correlation between groups $\mathbb{A}$ and $\mathbb{B}$: *e*

Types of Labellers (2)

- Average pair-wise correlation within group $\mathbb{A}$: c
- Average pair-wise correlation within group $\mathbb{B}$: d
- Average pair-wise correlation between groups $\mathbb{A}$ and $\mathbb{B}$: e
- Average of N labellers of group $\mathbb{A}$: A^N
- Average of N labellers of group $\mathbb{B}$: B^N

Types of Labellers (2)

- Average pair-wise correlation within group $\mathbb{A}$: c
- Average pair-wise correlation within group $\mathbb{B}$: d
- Average pair-wise correlation between groups $\mathbb{A}$ and $\mathbb{B}$: e
- Average of N labellers of group $\mathbb{A}$: A^N
- Average of N labellers of group $\mathbb{B}$: B^N
- **Quality** of A^N with respect to $\mathbb{B}$:

Types of Labellers (2)

- Average pair-wise correlation within group $\mathbb{A}$: c
- Average pair-wise correlation within group $\mathbb{B}$: d
- Average pair-wise correlation between groups $\mathbb{A}$ and $\mathbb{B}$: e
- Average of N labellers of group $\mathbb{A}$: A^N
- Average of N labellers of group $\mathbb{B}$: B^N
- **Quality** of A^N with respect to $\mathbb{B}$: Correlation to ground truth of $\mathbb{B}$

$$\mathrm{Corr}(A^N, B^\infty) = \frac{e}{\sqrt{\frac{1}{N} + \frac{N-1}{N} c} \cdot \sqrt{d}}$$

Types of Labellers (2)

- Average pair-wise correlation within group $\mathbb{A}$: c
- Average pair-wise correlation within group $\mathbb{B}$: d
- Average pair-wise correlation between groups $\mathbb{A}$ and $\mathbb{B}$: e
- Average of N labellers of group $\mathbb{A}$: A^N
- Average of N labellers of group $\mathbb{B}$: B^N
- **Quality** of A^N with respect to $\mathbb{B}$: Correlation to ground truth of $\mathbb{B}$

$$\mathrm{Corr}(A^N, B^\infty) = \frac{e}{\sqrt{\frac{1}{N} + \frac{N-1}{N} c} \cdot \sqrt{d}}$$

- Similarity between groups $\mathbb{A}$ and $\mathbb{B}$:

Types of Labellers (2)

- Average pair-wise correlation within group $\mathbb{A}$: c
- Average pair-wise correlation within group $\mathbb{B}$: d
- Average pair-wise correlation between groups $\mathbb{A}$ and $\mathbb{B}$: e
- Average of N labellers of group $\mathbb{A}$: A^N
- Average of N labellers of group $\mathbb{B}$: B^N
- **Quality** of A^N with respect to $\mathbb{B}$: Correlation to ground truth of $\mathbb{B}$

$$\mathrm{Corr}(A^N, B^\infty) = \frac{e}{\sqrt{\frac{1}{N} + \frac{N-1}{N} c \cdot \sqrt{d}}}$$

- Similarity between groups $\mathbb{A}$ and $\mathbb{B}$: Correlation of ground truths

$$\mathrm{Corr}(A^\infty, B^\infty) = \frac{e}{\sqrt{c \cdot d}}$$

Types of Labellers: Example

- Quality of rhythm on a 5-point Likert scale
- Compared three groups of labellers
  - Experts (Phoneticians with extensive labelling experience)
  - Phoneticians
  - Naïves

Types of Labellers: Example

- Quality of rhythm on a 5-point Likert scale
- Compared three groups of labellers
  - Experts (Phoneticians with extensive labelling experience)
  - Phoneticians
  - Naïves
- Average pairwise correlations:

	experts	phoneticians	naïves
experts	0.90		
phoneticians		0.81	
naïves			0.72

- When combining ≥ 5 labellers: Quality always ≥ 0.94
- Correlation of ground truths: ≥ 0.97

Types of Labellers: Example

- Quality of rhythm on a 5-point Likert scale
- Compared three groups of labellers
  - Experts (Phoneticians with extensive labelling experience)
  - Phoneticians
  - Naïves
- Average pairwise correlations:

	experts	phoneticians	naïves
experts	0.90	0.80	0.76
phoneticians		0.81	0.75
naïves			0.72

- When combining ≥ 5 labellers: Quality always ≥ 0.94
- Correlation of ground truths: ≥ 0.97

More Fun with Labellers

- Correlations between different annotated critertia
- Correlations when averaging over multiple annotated items

Humans versus Machines

- Pairwise correlation **underestimates** quality of average human
- Predicted scores Y of an automatic system trained with A^N?

Humans versus Machines

- Pairwise correlation **underestimates** quality of average human
- Predicted scores Y of an automatic system trained with A^N?
  - Quality: Correlation to ground truth

$$\mathrm{Corr}(Y, A^\infty) = \mathrm{Corr}(Y, A^N) \cdot \mathrm{Corr}(A^N, A^\infty)$$

Humans versus Machines

- Pairwise correlation **underestimates** quality of average human
- Predicted scores Y of an automatic system trained with A^N?
  - Quality: Correlation to ground truth

$$\mathrm{Corr}(Y, A^\infty) = \mathrm{Corr}(Y, A^N) \cdot \mathrm{Corr}(A^N, A^\infty)$$

  - Correlation to training labels **overestimates** quality of automatic system

Humans versus Machines

- Pairwise correlation **underestimates** quality of average human
- Predicted scores Y of an automatic system trained with A^N?
  - Quality: Correlation to ground truth

$$\mathrm{Corr}(Y, A^\infty) = \mathrm{Corr}(Y, A^N) \cdot \mathrm{Corr}(A^N, A^\infty)$$

  - Correlation to training labels **overestimates** quality of automatic system
- Example: 5 labellers, $c = 0.5$, correlation of Y to A^N: 0.6

Humans versus Machines

- Pairwise correlation **underestimates** quality of average human
- Predicted scores Y of an automatic system trained with A^N?
  - Quality: Correlation to ground truth

$$\mathrm{Corr}(Y, A^\infty) = \mathrm{Corr}(Y, A^N) \cdot \mathrm{Corr}(A^N, A^\infty)$$

  - Correlation to training labels **overestimates** quality of automatic system
- Example: 5 labellers, $c = 0.5$, correlation of Y to A^N: 0.6
- Quality of A^N: 0.91
- Quality of Y: $0.6 \cdot 0.91 = 0.55$

Humans versus Machines

- Pairwise correlation **underestimates** quality of average human
- Predicted scores Y of an automatic system trained with A^N?
  - Quality: Correlation to ground truth

$$\mathrm{Corr}(Y, A^\infty) = \mathrm{Corr}(Y, A^N) \cdot \mathrm{Corr}(A^N, A^\infty)$$

  - Correlation to training labels **overestimates** quality of automatic system
- Example: 5 labellers, $c = 0.5$, correlation of Y to A^N: 0.6
- Quality of A^N: 0.91
- Quality of Y: $0.6 \cdot 0.91 = 0.55$
- Quality of average human: $\sqrt{0.5} = 0.71$


BLAH

Automatic Assessment of Prosody

- Concrete phenomena for error detection:
  Word stress, phrase boundaries, phrase accents, tones, sentence mood
- Rhythm/Intonation/Sentence Melody?

Automatic Assessment of Prosody

- Concrete phenomena for error detection:
  Word stress, phrase boundaries, phrase accents, tones, sentence mood
- Rhythm/Intonation/Sentence Melody?

- Perceptual ratings
  - THIS SENTENCE'S MELODY SOUNDS…
    (1) *normal* (2) *acceptable, but not perfectly normal* (3) *slightly unusual*
    (4) *unusual* (5) *very unusual*
  - THE ENGLISH LANGUAGE HAS A CHARACTERISTIC RHYTHM (TIMING OF THE SYLLABLES). HOW DO YOU ASSESS THE RHYTHM OF THIS SENTENCE?
    (1) *normal* (2) *acceptable, but not perfectly normal* (3) *slightly unusual*
    (4) *unusual* (5) *very unusual*
- Predict continuous scores with regression system (on utterance level)

Generative versus Discriminative Approaches

- For evaluation: Always need (some) annotated non-native data
- Modelling: Two basic approaches

Generative versus Discriminative Approaches

- For evaluation: Always need (some) annotated non-native data
- Modelling: Two basic approaches
- Generative or **indirect**:
  - Model describes what is acceptable
  - Distance measure used for classification or as continuous score
  + Data collection far easier
  - Example: GOP

Generative versus Discriminative Approaches

- For evaluation: Always need (some) annotated non-native data
- Modelling: Two basic approaches
- Generative or **indirect**:
  - Model describes what is acceptable
  - Distance measure used for classification or as continuous score
  - \+ Data collection far easier
  - Example: GOP
- Discriminative or **direct**:
  - Model both acceptable and unacceptable pronunciations
  - Score or decision ‘correct’ or ‘wrong’ is direct output of classifier / regression system
  - \+ Potential for optimal accuracy
  - − Much more, more expensive data needed
  - Example: Individual phoneme error patterns


STOP/EJECT
PLAY
FORWARD
REWIND
RECORD
Fisher-Price

C-AuDiT (English Part)

- Read material, about 30 hours
- 25 German, 10 French, 10 Spanish, 10 Italian, and 2 Hindi speakers
- 329 utterances: Two short stories, tongue twisters, sentences with different position of phrase accents, shift of word accent

C-AuDiT (English Part)

- Read material, about 30 hours
- 25 German, 10 French, 10 Spanish, 10 Italian, and 2 Hindi speakers
- 329 utterances: Two short stories, tongue twisters, sentences with different position of phrase accents, shift of word accent
- Detailled annotation by 3 experts
- Subset of 5 sentences
  - Perceptual annotation: Intelligibility, non-native accent, sentence melody and rhythm on a 5-point Likert scale
  - 20 American, 19 British, 21 Scottish naïves
  - Included data from ISLE database $\rightarrow$ approx. 1 hour from 94 speakers

C-AuDiT (English Part)

- Read material, about 30 hours
- 25 German, 10 French, 10 Spanish, 10 Italian, and 2 Hindi speakers
- 329 utterances: Two short stories, tongue twisters, sentences with different position of phrase accents, shift of word accent
- Detailled annotation by 3 experts
- Subset of 5 sentences
  - Perceptual annotation: Intelligibility, non-native accent, sentence melody and rhythm on a 5-point Likert scale
  - 20 American, 19 British, 21 Scottish naïves
  - Included data from ISLE database $\rightarrow$ approx. 1 hour from 94 speakers
- Difference between American/British/Scottish only for accent
  $\rightarrow$ combine all
- Rhythm and melody scores highly correlated: 0.95 (ground truths: 0.97)
  $\rightarrow$ combine into one score **pros**
- Quality: 0.99

Dialogue of the Day (dod)

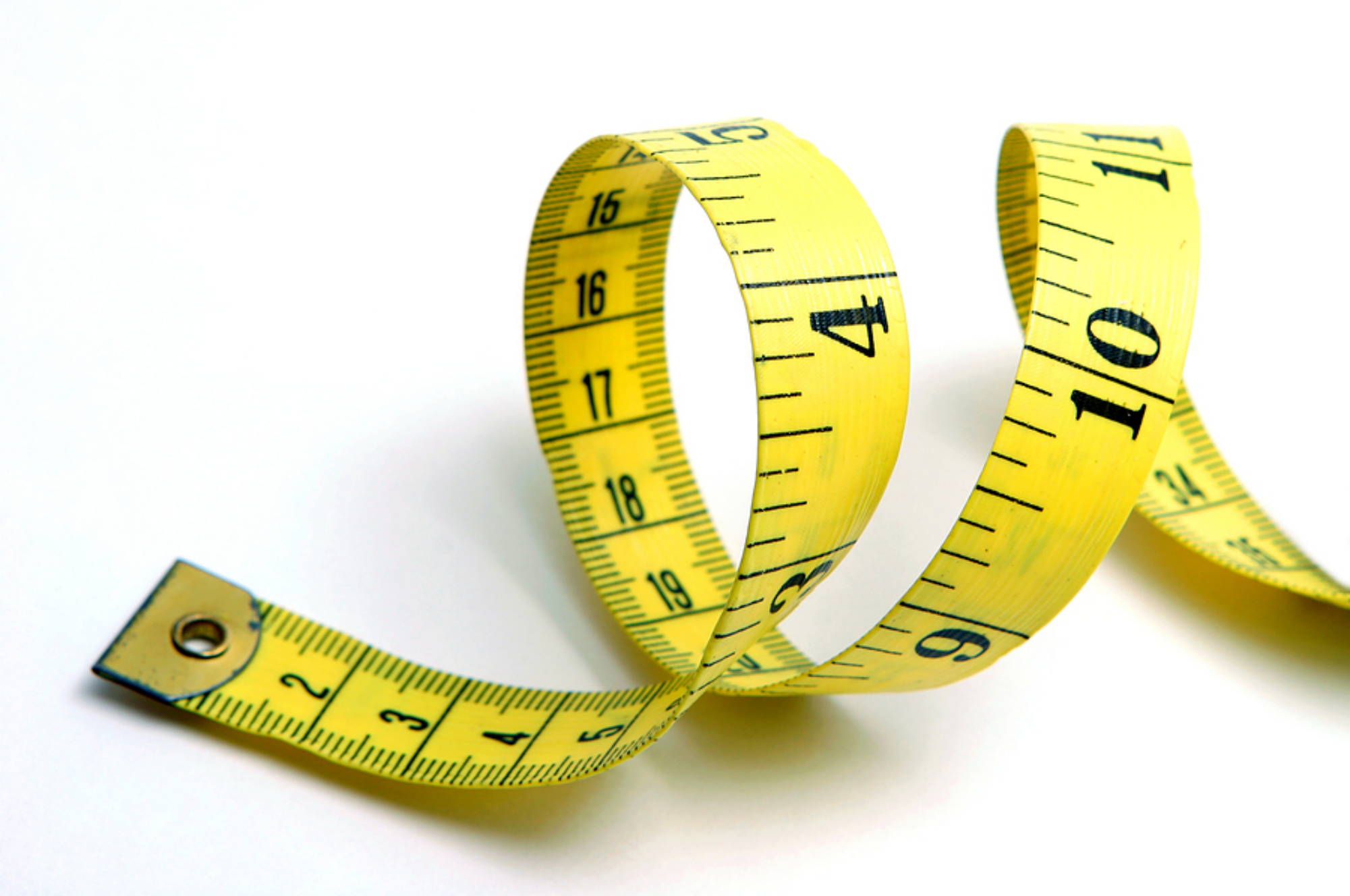
- Web-based tool for training pre-scripted dialogues
- Different enacting modes
  - Just listen
  - Repeat own lines after reference speaker/read off screen
  - With/without shadowing by reference speaker
  - Slow mode
  - Subdivide longer utterances

Dialogue of the Day (dod)

- Web-based tool for training pre-scripted dialogues
- Different enacting modes
  - Just listen
  - Repeat own lines after reference speaker/read off screen
  - With/without shadowing by reference speaker
  - Slow mode
  - Subdivide longer utterances
- 18 dialogues on business negotiations, shopping, etc.
- 6 professional reference speakers in normal and slow tempo
- Semi-spontaneous
- 85 volunteering learners got login; usable material from 31 speakers

Dialogue of the Day (dod)

- Web-based tool for training pre-scripted dialogues
- Different enacting modes
  - Just listen
  - Repeat own lines after reference speaker/read off screen
  - With/without shadowing by reference speaker
  - Slow mode
  - Subdivide longer utterances
- 18 dialogues on business negotiations, shopping, etc.
- 6 professional reference speakers in normal and slow tempo
- Semi-spontaneous
- 85 volunteering learners got login; usable material from 31 speakers
- Classification into clean (5.5h) usable (1.7h) and unusable (0.6h) speech
- Perceptually annotated by 5 phoneticians
- Quality of **pros** labels: 0.85



Prosodic Features

- Input parameters for regression system
- Prosodic ‘fingerprint’ of the utterance
- Fully automatic
- Assumption: Target utterance has indeed been spoken
- Segmentation via forced alignment

Specialized Rhythm Features

- Body of research on modelling rhythm of L1s
- Specialized, hand-crafted parameters; promising for our task
- Duration Features **Dur**:
  $\varnothing$ duration of syllables, $\varnothing$ duration of vocalic and consonantal intervals (2 features)
- Isochrony Features **Iso**:
  Distances between consecutive stressed or unstressed syllables ($\varnothing$, σ, ratios: 6 features)
- Pairwise Variability Indices **PVI** (Grabe and Low; Bertinetto and Bertini):
  Absolute difference in duration of consecutive intervals (vocalic, consonantal, raw, normalized: 4 features);
  Control/compensation index (CCI): 2 features

Specialized Rhythm Features (2)

- Global Interval Proportions **GPI** (Ramus; Dellwo):
  % of vocalic intervals;
  'Deltas': σ of duration of vocalic and consonantal intervals;
  variation coefficients 'Varco' (normalized Deltas),
  (in total 5 features)
- Combination the above: **Rhy-All** (19 features)

General-Purpose-Features

- Capture as much as possible potentially relevant prosodic information
- Somewhere between knowledge-based and brute-force
- Local Features
  - Our veteran general purpose ‘prosody module’
  - Based on duration, energy, pitch, and pauses
  - Characterizes an arbitrary unit of speech
  - Energy and F0 are suitably preprocessed and perceptually transformed
  - Handful of functionals: Max, max-pos, slope . . .
  - Normalized versions account for phoneme-intrinsic variations

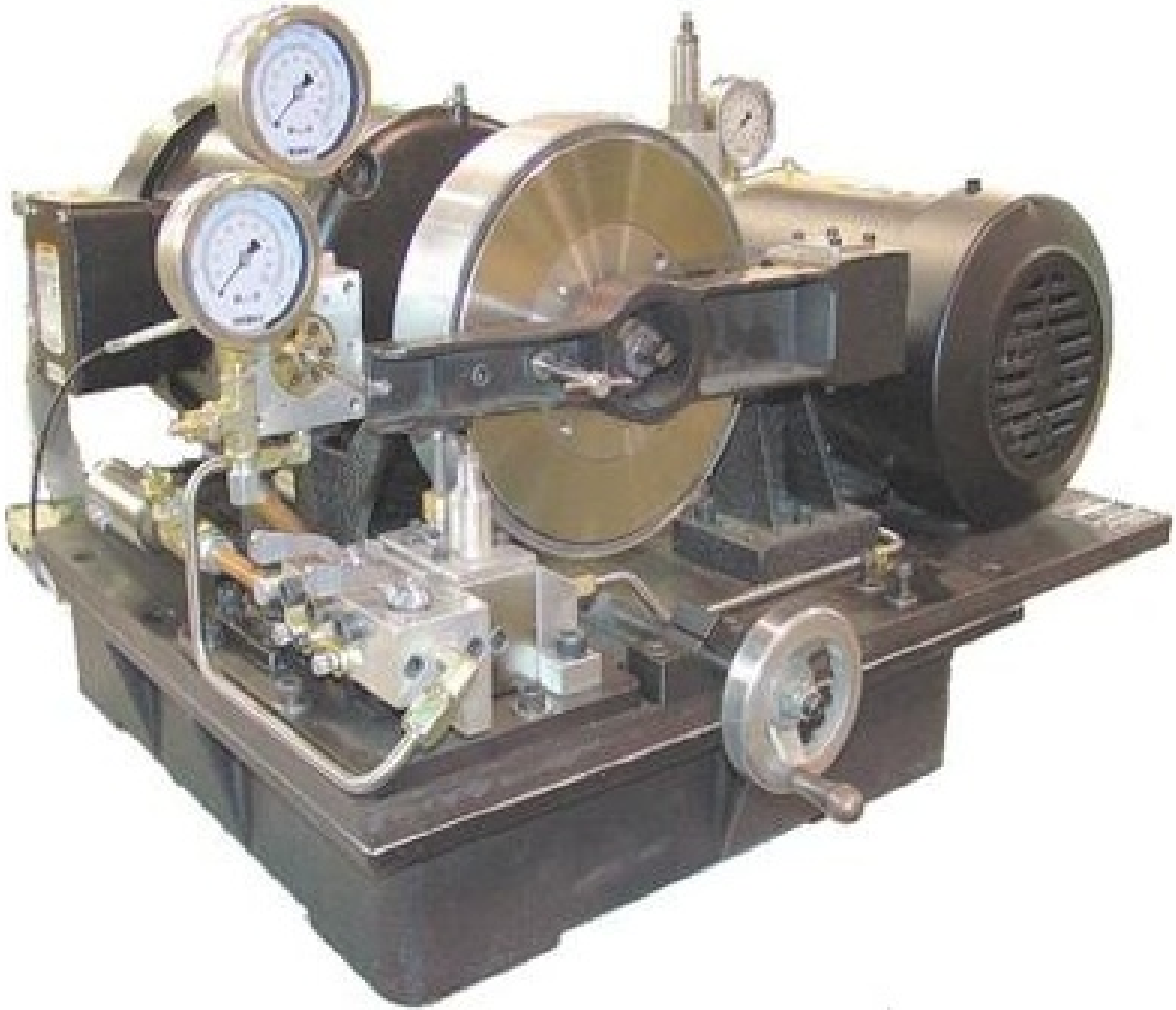
General-Purpose-Features: Globally

- Apply module to different units and derive utterance-level features, as exhaustively as possible (742 features **Pros**)
  - $\varnothing$ and σ of the local features of all **stressed syllables** ± 2 syllables context (same for just the nuclei of stressed syllables)
  - $\varnothing$ and σ of the local features of all **words** and **consecutive pairs of words** (same for syllables and nuclei)
  - $\varnothing$ abs. difference between the local features of **consecutive words** (same for syllables and nuclei)

General-Purpose-Features: Globally

- Apply module to different units and derive utterance-level features, as exhaustively as possible (742 features **Pros**)
  - $\varnothing$ and σ of the local features of all **stressed syllables** ± 2 syllables context (same for just the nuclei of stressed syllables)
    (isochrony)
  - $\varnothing$ and σ of the local features of all **words** and **consecutive pairs of words** (same for syllables and nuclei)
    (proportions and deltas)
  - $\varnothing$ abs. difference between the local features of **consecutive words** (same for syllables and nuclei)
    (pairwise variability indices)

- Extension **All**: Append **Rhy-All**



Global Scoring Model

- Take a feature set as input for a regression system to directly predict sentence score
- Support Vector Regression (SVR)

Local Scoring Model

- Divide-and-conquer strategy
  1. Score all **syllables** individually by SVR
  2. Combine by averaging
- Syllable label: Just the utterance label
- Features
  - Local general purpose prosodic features for current syllable, nucleus and word, ± 2 context (312 features)
  - Word accents and (prototypical) boundaries in a neighbourhood of ± 2 syllables, position within word and utterance (60 features)
  - Phrase accent, number of syllables, position in utterance etc. of current word (10 features)
  - Number of words, sentence mood (4 features)

Local Scoring Model (2)

- Aims when developing **Local**
  - Higher robustness (more training data, averaging)
  - Capture more information (chronological, utterance structure)
- Extensions **Local+Pros**, **Local+All**:
  Enrich each syllable's feature vector with global features

Nuisance Removal

- Obtaining representative data is difficult
- Improve by applying prior knowledge
- Possible invariance for rhythm: Tempo
  - Pronunciation scores should be independent of tempo
  - Good learners tend to speak faster

Nuisance Removal (2)

- Estimate tempo T by $\varnothing$ syllable duration
- Modify Reference X: Remove correlation to T

$$X' := X - \frac{\mathrm{Cov}(X, T)}{\mathrm{Var}(T)} \cdot T$$

- C-AuDiT
  - T correlated strongly with **pros** (0.59)
  - Quality suffers not much (0.99 $\rightarrow$ 0.98).
- dod
  - T correlated less with **pros** (0.23)
  - Quality suffered a bit (0.85 $\rightarrow$ 0.80).

Evaluation Schemes: Depend on Application

- Language Testing
  - Draw test items from **fixed, known set**
  - Special tailoring of test items
  - Calibration of items
  - Global scores
  - Feasible & established

Evaluation Schemes: Depend on Application

- Language Testing
  - Draw test items from **fixed, known set**
  - Special tailoring of test items
  - Calibration of items
  - Global scores
  - Feasible & established

- CAPT
  - Must be applicable to wide range of speech items
  - Prompts **unknown** when developing system
  - Local scores wanted
  - Tricky

Evaluation Schemes: Depend on Application

- Language Testing
  - Draw test items from **fixed, known set**
  - Special tailoring of test items
  - Calibration of items
  - Global scores
  - Feasible & established
  - Evaluation: Test on unseen speakers
- CAPT
  - Must be applicable to wide range of speech items
  - Prompts **unknown** when developing system
  - Local scores wanted
  - Tricky

Evaluation Schemes: Depend on Application

- Language Testing
  - Draw test items from **fixed, known set**
  - Special tailoring of test items
  - Calibration of items
  - Global scores
  - Feasible & established
  - Evaluation: Test on unseen speakers
- CAPT
  - Must be applicable to wide range of speech items
  - Prompts **unknown** when developing system
  - Local scores wanted
  - Tricky
  - Evaluation: Test on unseen speakers **and** unseen speech items

Evaluation Schemes

- **Speaker-independent** cross-validation (CV)
  - In each fold, train and test is disjunct w.r.t speakers
  - 2 folds
  - 50% can be used for train and test, respectively
- **Speaker- and utterance-independent** CV
  - Train and test is disjunct w.r.t speakers **and** utterances
  - C-AuDiT: 5-fold utterance CV within a 2-fold speaker CV
    $2 \times 5 = 10$ folds in total
    $4/5 \times 1/2 = 40\%$ of the data can be used for training
    $1/5 \times 1/2 = 10\%$ for testing
    50% cannot be used
  - dod: Two-fold utterance CV within 2-fold speaker CV
    $2 \times 2 = 4$ folds in total
    $1/2 \times 1/2 = 25\%$ of the data can be used for training and testing, resp.
    50% cannot be used

Meta-Parameters

- Vital to choose suitable parameters for SVR
- 3 Kernels: 1 Linear, 2 non-linear
- 4 Values for complexity parameter *C*
- For each feature set, take the combination that works best in the CV
- Optimization on **test**, but bias should be limited

Results

Corpus	Evaluation	Reference	**Dur**	**Iso**	**PVI**	**GPI**	**Rhy-All**	**Pros**	**All**	**Local**	**Local +Pros**	**Local +All**
C-AuDiT	Speaker	Orig	0.60	0.58	0.59	0.45	0.73	**0.75**	**0.75**	0.69	**0.75**	**0.75**
		w/o Dur	0.14	0.22	0.45	0.34	0.54	0.58	0.58	0.50	**0.59**	**0.59**
	Spk+Sent	Orig	0.54	0.50	0.42	0.19	0.58	0.53	0.53	**0.64**	0.54	0.54
		w/o Dur	-0.13	-0.15	0.25	0.16	0.33	0.26	0.28	0.21	0.33	**0.34**
dod	Speaker	Orig	0.48	0.50	0.41	0.37	0.56	**0.57**	**0.57**	0.53	**0.57**	**0.57**
		w/o Dur	0.40	0.43	0.38	0.34	0.50	0.52	0.52	0.50	**0.53**	**0.53**
	Spk+Sent	Orig	0.40	0.45	0.33	0.31	0.49	**0.52**	**0.52**	0.48	0.51	**0.52**
		w/o Dur	0.32	0.37	0.31	0.28	0.42	0.47	0.47	0.44	0.47	**0.48**

Results

Corpus	Evaluation	Reference	**Dur**	**Iso**	**PVI**	**GPI**	**Rhy-All**	**Pros**	**All**	**Local**	**Local +Pros**	**Local +All**
C-AuDiT	Speaker	Orig	0.60	0.58	0.59	0.45	0.73	**0.75**	**0.75**	0.69	**0.75**	**0.75**

Results

Corpus	Evaluation	Reference	**Dur**	**Iso**	**PVI**	**GPI**	**Rhy-All**	**Pros**	**All**	**Local**	**Local +Pros**	**Local +All**
C-AuDiT	Speaker	Orig	0.60	0.58	0.59	0.45	0.73	**0.75**	**0.75**	0.69	**0.75**	**0.75**
		w/o Dur	0.14	0.22	0.45	0.34	0.54	0.58	0.58	0.50	**0.59**	**0.59**

Results

Corpus	Evaluation	Reference	**Dur**	**Iso**	**PVI**	**GPI**	**Rhy-All**	**Pros**	**All**	**Local**	**Local +Pros**	**Local +All**
C-AuDiT	Speaker	Orig	0.60	0.58	0.59	0.45	0.73	**0.75**	**0.75**	0.69	**0.75**	**0.75**
		w/o Dur	0.14	0.22	0.45	0.34	0.54	0.58	0.58	0.50	**0.59**	**0.59**
	Spk+Sent	Orig	0.54	0.50	0.42	0.19	0.58	0.53	0.53	**0.64**	0.54	0.54
		w/o Dur	-0.13	-0.15	0.25	0.16	0.33	0.26	0.28	0.21	0.33	**0.34**

Results

Corpus	Evaluation	Reference	**Dur**	**Iso**	**PVI**	**GPI**	**Rhy-All**	**Pros**	**All**	**Local**	**Local +Pros**	**Local +All**
C-AuDiT	Speaker	Orig	0.60	0.58	0.59	0.45	0.73	**0.75**	**0.75**	0.69	**0.75**	**0.75**
		w/o Dur	0.14	0.22	0.45	0.34	0.54	0.58	0.58	0.50	**0.59**	**0.59**
	Spk+Sent	Orig	0.54	0.50	0.42	0.19	0.58	0.53	0.53	**0.64**	0.54	0.54
		w/o Dur	-0.13	-0.15	0.25	0.16	0.33	0.26	0.28	0.21	0.33	**0.34**
dod	Speaker	Orig	0.48	0.50	0.41	0.37	0.56	**0.57**	**0.57**	0.53	**0.57**	**0.57**
		w/o Dur	0.40	0.43	0.38	0.34	0.50	0.52	0.52	0.50	**0.53**	**0.53**
	Spk+Sent	Orig	0.40	0.45	0.33	0.31	0.49	**0.52**	**0.52**	0.48	0.51	**0.52**
		w/o Dur	0.32	0.37	0.31	0.28	0.42	0.47	0.47	0.44	0.47	**0.48**

Results

Corpus	Evaluation	Reference	**Dur**	**Iso**	**PVI**	**GPI**	**Rhy-All**	**Pros**	**All**	**Local**	**Local +Pros**	**Local +All**
C-AuDiT	Speaker	Orig	0.60	0.58	0.59	0.45	0.73	**0.75**	**0.75**	0.69	**0.75**	**0.75**
		w/o Dur	0.14	0.22	0.45	0.34	0.54	0.58	0.58	0.50	**0.59**	**0.59**
	Spk+Sent	Orig	0.54	0.50	0.42	0.19	0.58	0.53	0.53	**0.64**	0.54	0.54
		w/o Dur	-0.13	-0.15	0.25	0.16	0.33	0.26	0.28	0.21	0.33	**0.34**
dod	Speaker	Orig	0.48	0.50	0.41	0.37	0.56	**0.57**	**0.57**	0.53	**0.57**	**0.57**
		w/o Dur	0.40	0.43	0.38	0.34	0.50	0.52	0.52	0.50	**0.53**	**0.53**
	Spk+Sent	Orig	0.40	0.45	0.33	0.31	0.49	**0.52**	**0.52**	0.48	0.51	**0.52**
		w/o Dur	0.32	0.37	0.31	0.28	0.42	0.47	0.47	0.44	0.47	**0.48**

- For most of the application-relevant test setups (Reference: Orig), **Rhy-All** does surprisingly well
- Modelling power (w/o Dur): More complex features clearly ahead

Discussion

- Sentence-dependent results for C-AuDiT quite good
  - Applicability for CAPT limited
  - For applications with fixed inventory, other methods may be cheaper and even better
- Best sentence-independent result for C-AuDiT (late fusion of **Pros** and **Local**, not in table): 0.67
  Quality: $0.67 \cdot 0.99 = 0.66$ clearly ahead of avg. labeller (0.54)
- For dod: $0.53 \cdot 0.85 = 0.45$, clearly behind of avg. labeller (0.58)

Conclusion

- Question your machine learning results
- Try to evaluate as realistically as possible
- Anticipate evaluation when collecting data
- Scoring general prosody on a continuous scale on utterance level, playing in the same league as an average human
- Correlations improve quickly when averaging over multiple utterances

Conclusion

- Question your machine learning results
- Try to evaluate as realistically as possible
- Anticipate evaluation when collecting data
- Scoring general prosody on a continuous scale on utterance level, playing in the same league as an average human
- Correlations improve quickly when averaging over multiple utterances

- Is adept?

Questions?

Removing Nuisance

- $\boldsymbol{z} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix}, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{y} \in \mathbb{R}^M$
- $\boldsymbol{x}$: “nuisance”
- Goal: Remove from y_j any correlation to x_i
- $\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{E}^T \\ \boldsymbol{E} & \boldsymbol{D} \end{pmatrix}, \quad \boldsymbol{C} \in \mathbb{R}^{NxN}, \boldsymbol{E} \in \mathbb{R}^{MxN}, \boldsymbol{D} \in \mathbb{R}^{MxM}$
- $\boldsymbol{y}' = \boldsymbol{T}\boldsymbol{z}$ with
  $\boldsymbol{T} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{I_M} \end{pmatrix}$ and
  $\boldsymbol{AC} = -\boldsymbol{E}$
- First approach: Avg. syll. duration: $\boldsymbol{x}$, features: $\boldsymbol{y}$

Removing Nuisance

- $\boldsymbol{z} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix}, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{y} \in \mathbb{R}^M$
- $\boldsymbol{x}$: “nuisance”
- Goal: Remove from y_j any correlation to x_i
- $\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{E}^T \\ \boldsymbol{E} & \boldsymbol{D} \end{pmatrix}, \quad \boldsymbol{C} \in \mathbb{R}^{NxN}, \boldsymbol{E} \in \mathbb{R}^{MxN}, \boldsymbol{D} \in \mathbb{R}^{MxM}$
- $\boldsymbol{y}' = \boldsymbol{T}\boldsymbol{z}$ with
  $\boldsymbol{T} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{I_M} \end{pmatrix}$ and
  $\boldsymbol{AC} = -\boldsymbol{E}$
- ~~First approach: Avg. syll. duration: $\boldsymbol{x}$, features: $\boldsymbol{y}$~~
- Second approach: avg. syll. duration: $\boldsymbol{x}$, target value: $\boldsymbol{y}$