Crowd Translator: On Building Localized Speech Recognizers through Micropayments

Nokia Research Center, Cambridge US

Jonathan Ledlie, Billy Odero, Einat Minkov, Imre Kiss, Joseph Polifroni

October 2009

NOKIA

Overview

Overview

Scenario: mobile money transfer

- MPESA in Kenya
- Current UI is text based
  - Literacy is major barrier
  - Voice-based UI had much higher task completion rates (Medhi, CHI '09)

Overview

Scenario: mobile money transfer

- MPESA in Kenya
- Current UI is text based
  - Literacy is major barrier
  - Voice-based UI had much higher task completion rates (Medhi, CHI '09)

Purchase speech recognizer?

- "Rich country" languages only
  - Gaelic (0.5m), Welsh (1m)
- Out of luck
  - Luo (3.5m), Swahili (5m 1st, 80m 2nd)

Overview

Scenario: mobile money transfer

- MPESA in Kenya
- Current UI is text based
  - Literacy is major barrier
  - Voice-based UI had much higher task completion rates (Medhi, CHI '09)

Purchase speech recognizer?

- "Rich country" languages only
  - Gaelic (0.5m), Welsh (1m)
- Out of luck
  - Luo (3.5m), Swahili (5m 1st, 80m 2nd)

Our Approach: Crowd Translator

- Cheaply create recognizer for local, low-corpus languages

How is a speech recognizer built?

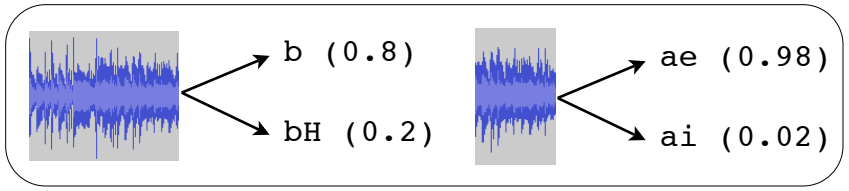
Type 1: PHONEME

1. Expert creates dictionary

```
bangers    b-ae-N-s@r-z
batter     b-ae-t-s@r
...        ...
```

2. Collect corpus from native speakers

3. Build phoneme matcher




```
Output: bangers (98%)
        batter   ( 1%)
        ...
```

Type 2: PHRASE

How is a speech recognizer built?

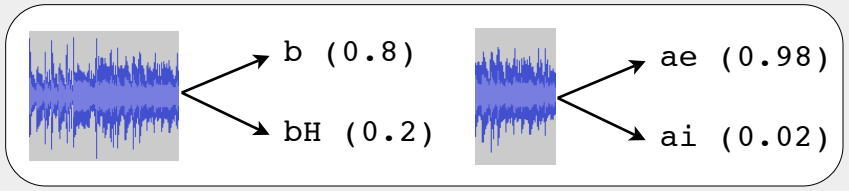
Type 1: PHONEME

1. Expert creates dictionary

```
bangers    b-ae-N-s@r-z
batter     b-ae-t-s@r
...        ...
```

2. Collect corpus from native speakers

3. Build phoneme matcher




```
Output: bangers (98%)
        batter   ( 1%)
        ...
```

Type 2: PHRASE

How is a speech recognizer built?

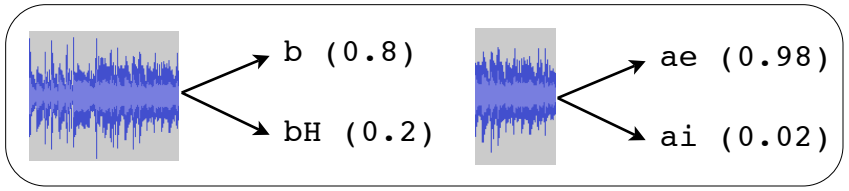
Type 1: PHONEME

1. Expert creates dictionary

```
bangers    b-ae-N-s@r-z
batter     b-ae-t-s@r
...        ...
```

2. Collect corpus from native speakers

3. Build phoneme matcher



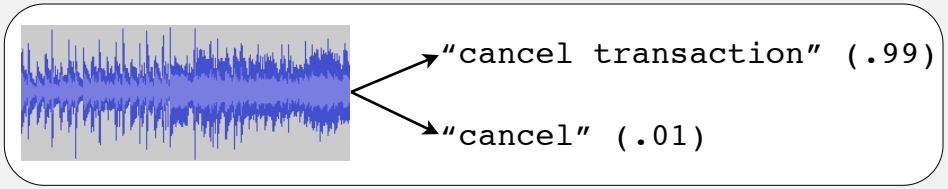

```
Output: bangers (98%)
        batter   ( 1%)
        ...
```

Type 2: PHRASE

1. Determine target phrases (no expert)
2. Collect corpus from native speakers

```
"new transaction"
"cancel transaction"
"cancel"
```

3. Build phrase matcher




```
Output: cancel transaction (99%)
        cancel (1%)
```


NOKIA
Connecting People

How is a speech recognizer built?

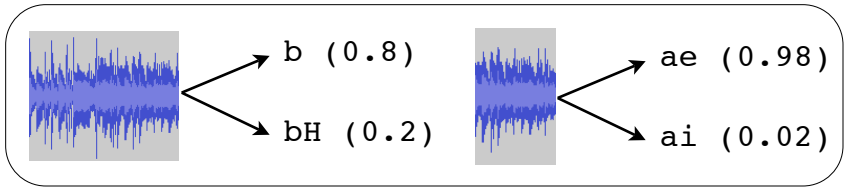
Type 1: PHONEME

1. Expert creates dictionary

```
bangers    b-ae-N-s@r-z
batter     b-ae-t-s@r
...        ...
```

2. Collect corpus from native speakers

3. Build phoneme matcher




```
Output: bangers (98%)
        batter   ( 1%)
        ...
```

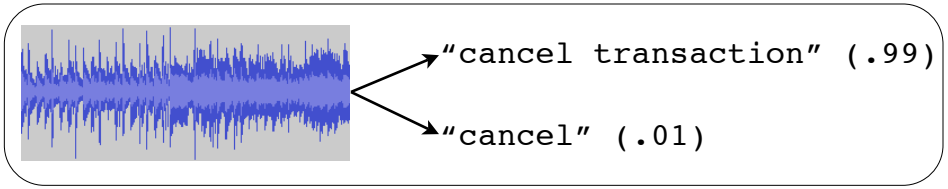
- Expensive (>$10m/language)
+ Grammar expandable
+ Memory: |phonemes|

Type 2: PHRASE

1. Determine target phrases (no expert)
2. Collect corpus from native speakers

```
"new transaction"
"cancel transaction"
"cancel"
```

3. Build phrase matcher




```
Output: cancel transaction (99%)
        cancel (1%)
```

+ Cheap ($10k/language)
- Corpus not expandable without more collection
~ Memory: |vocabulary|
  Good enough for C&C on devices w/small vocab

NOKIA
Connecting People

How is a speech recognizer built?

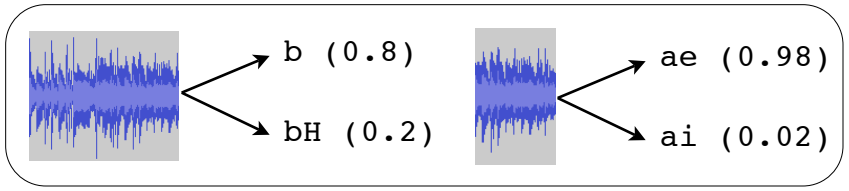
Type 1: PHONEME

1. Expert creates dictionary

```
bangers    b-ae-N-s@r-z
batter     b-ae-t-s@r
...        ...
```

2. Collect corpus from native speakers

3. Build phoneme matcher




```
Output: bangers (98%)
        batter   ( 1%)
        ...
```

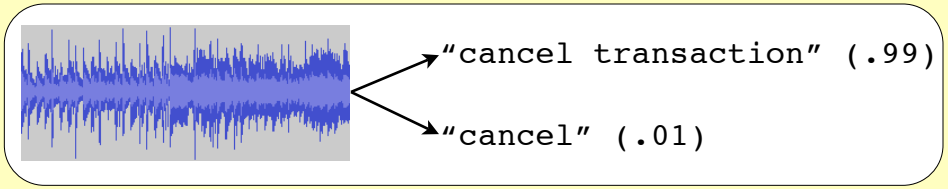
- Expensive (>$10m/language)
+ Grammar expandable
+ Memory: |phonemes|

Type 2: PHRASE

1. Determine target phrases (no expert)
2. Collect corpus from native speakers

```
"new transaction"
"cancel transaction"
"cancel"
```

3. Build phrase matcher




```
Output: cancel transaction (99%)
        cancel (1%)
```

+ Cheap ($10k/language)
- Corpus not expandable without more collection
~ Memory: |vocabulary|
  Good enough for C&C on devices w/small vocab

NOKIA
Connecting People

CX Design Goals

- Gather large corpus from native speakers
- Establish user trust
- Keep total costs low

Our Approach

(a) Make Canonical Recordings

English	Swahili	Gold Std. Utterance
car	gari	“gari”
boat	mashua	“mashua”
plane	ndege	“ndege”
…	…	“…”

(c) Verify Input

(b) Gather User Input

(d) Expand Corpus

Our Approach

(a) Make Canonical Recordings

English	Swahili	Gold Std. Utterance
car	gari	"gari"
boat	mashua	"mashua"
plane	ndege	"ndege"
…	…	"…"

(b) Gather User Input

Prompt	$User_1$ Utterance
$gari_g$	$gari_1$
$ndege_g$	$ndege_1$
$mashua_g$	$mashua_1$
…g	…1
$gari_g$	$gari_1'$
…g	…1
$mashua_g$	$mashua_1'$

(c) Verify Input

(d) Expand Corpus

NOKIA
Connecting People

Our Approach

(a) Make Canonical Recordings

English	Swahili	Gold Std. Utterance
car	gari	"gari"
boat	mashua	"mashua"
plane	ndege	"ndege"
…	…	"…"

(b) Gather User Input

Prompt	$User_1$ Utterance
$gari_g$	$gari_1$
$ndege_g$	$ndege_1$
$mashua_g$	$mashua_1$
…g	…1
$gari_g$	$gari_1'$
…g	…1
$mashua_g$	$mashua_1'$

(c) Verify Input

Intra-session Agreement?

$gari_1 \approx gari_1'$

$mashua_1 \approx mashua_1'$

(d) Expand Corpus

Our Approach

(a) Make Canonical Recordings

English	Swahili	Gold Std. Utterance
car	gari	"gari"
boat	mashua	"mashua"
plane	ndege	"ndege"
…	…	"…"

(b) Gather User Input

Prompt	$User_1$ Utterance
$gari_g$	$gari_1$
$ndege_g$	$ndege_1$
$mashua_g$	$mashua_1$
…g	…1
$gari_g$	$gari_1'$
…g	…1
$mashua_g$	$mashua_1'$

(c) Verify Input

Intra-session Agreement?

$gari_1 \approx gari_1'$

$mashua_1 \approx mashua_1'$

(d) Expand Corpus

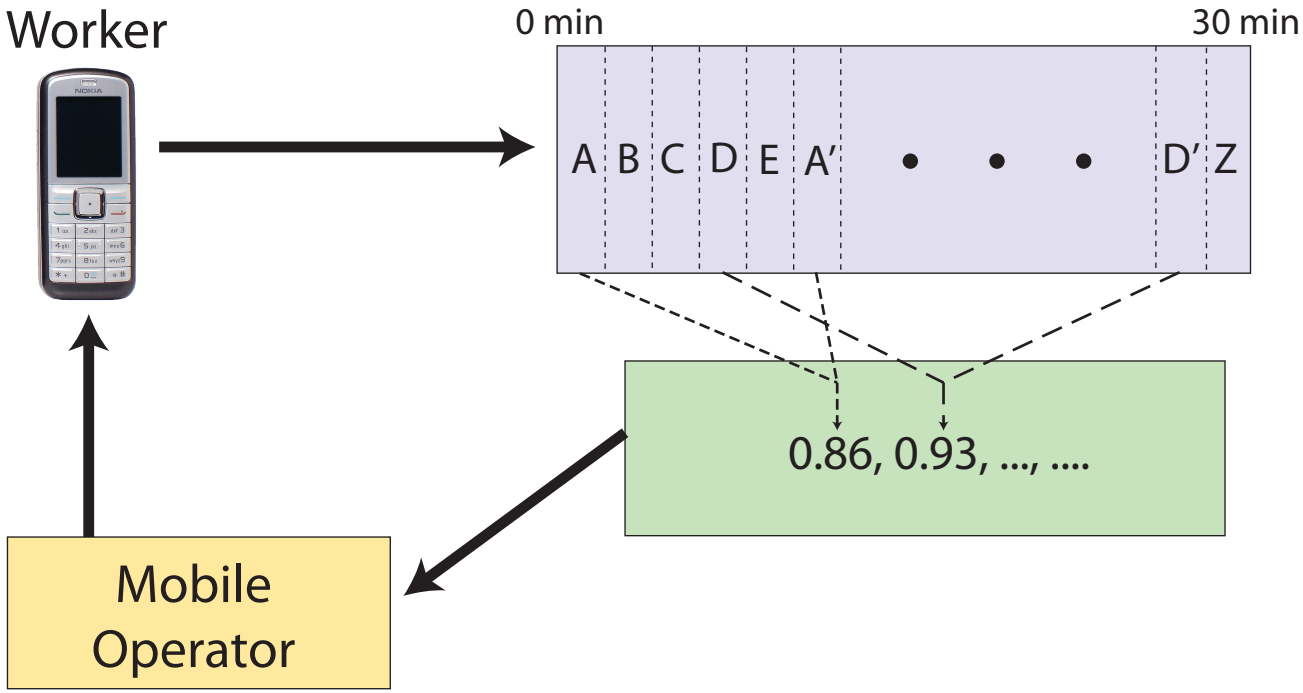
Word	Utterance
car	$gari_g$
car	$gari_1$
car	$gari_1'$
car	$gari_2$
…	…
boat	$mashua_g$
boat	$mashua_1$
…	…

Added

NOKIA
Connecting People

System Overview

Like Mechanical Turk: pay users for validated work (i.e. speech contributions)

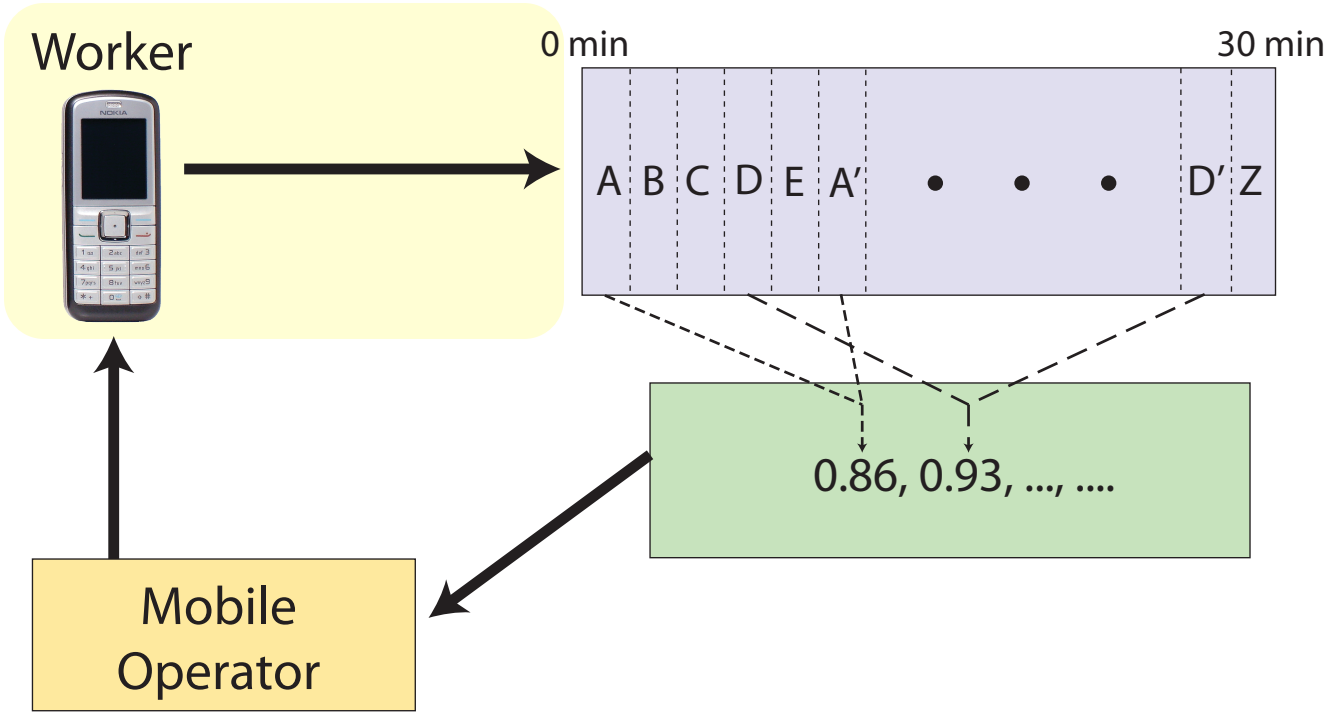


NOKIA
Connecting People

System Overview

Like Mechanical Turk: pay users for validated work (i.e. speech contributions)

1. User flashes CX
   - Gets callback

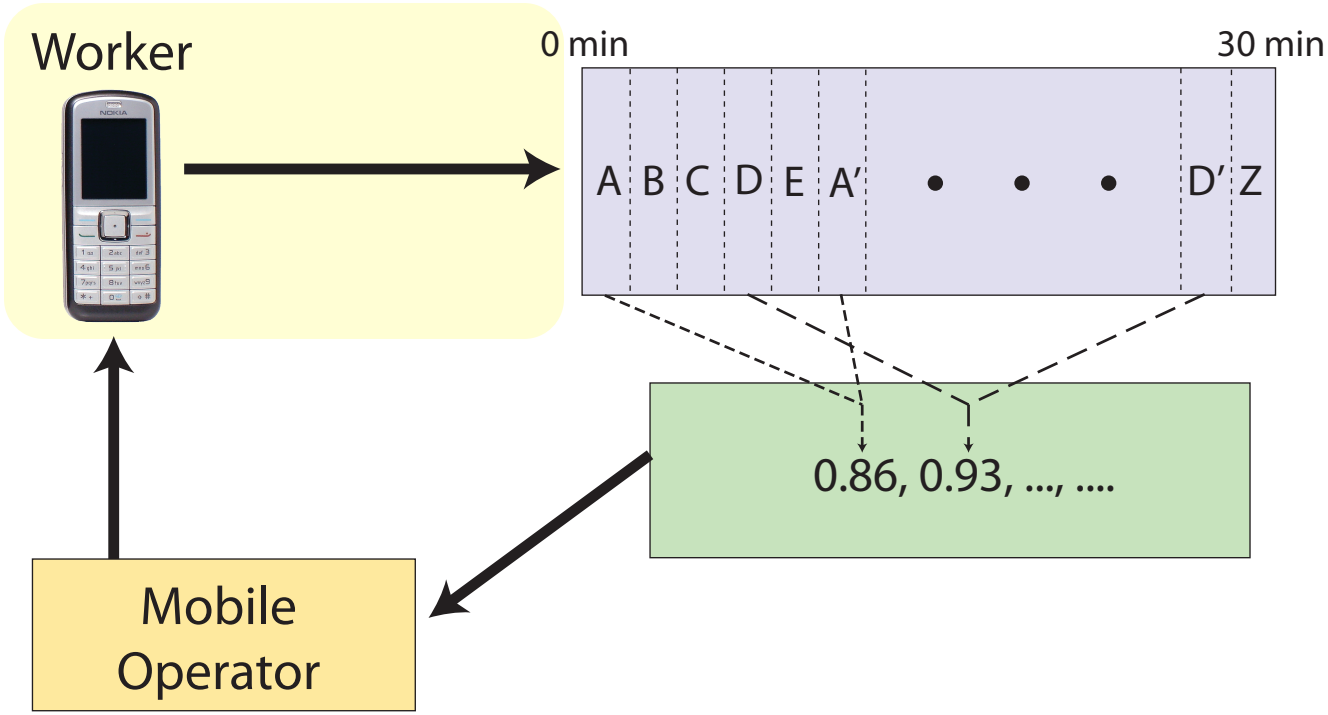


NOKIA
Connecting People

System Overview

Like Mechanical Turk: pay users for validated work (i.e. speech contributions)

1. User flashes CX
   - Gets callback
2. Selects his native language

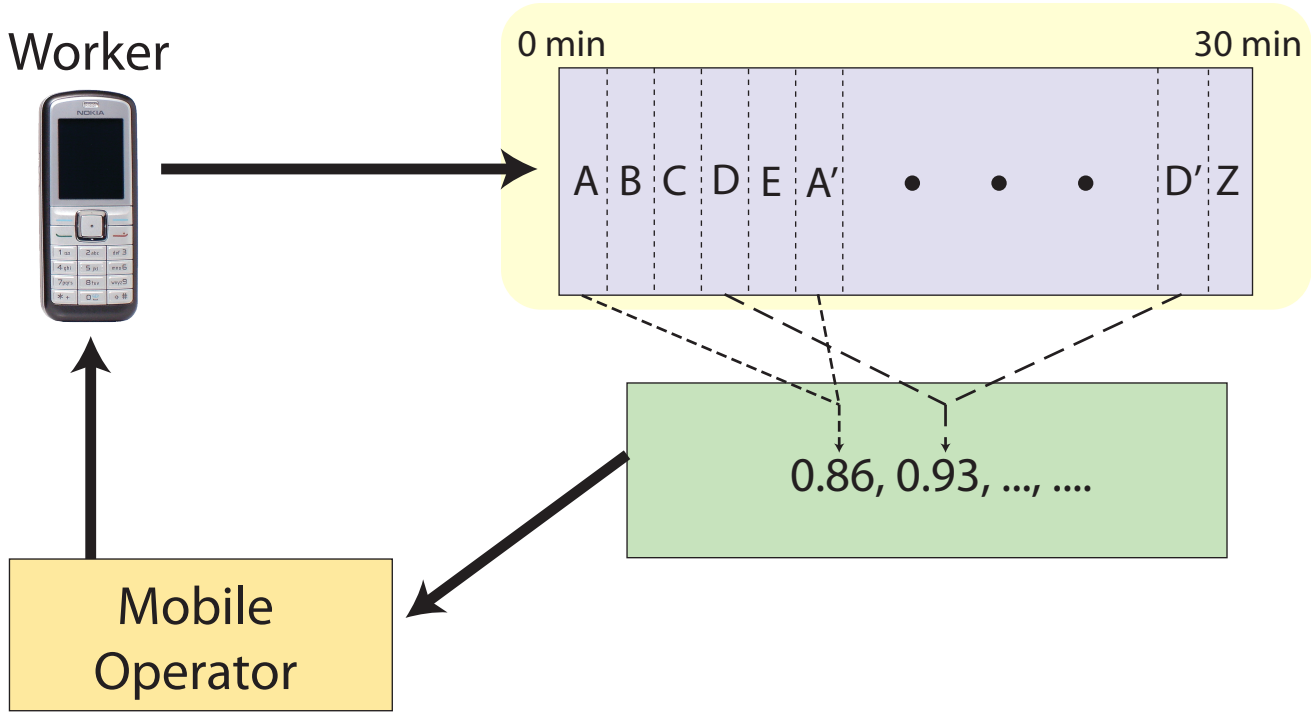


NOKIA
Connecting People

System Overview

Like Mechanical Turk: pay users for validated work (i.e. speech contributions)

1. User flashes CX
   - Gets callback
2. Selects his native language
3. Mimics voice prompts
   - "new transaction"

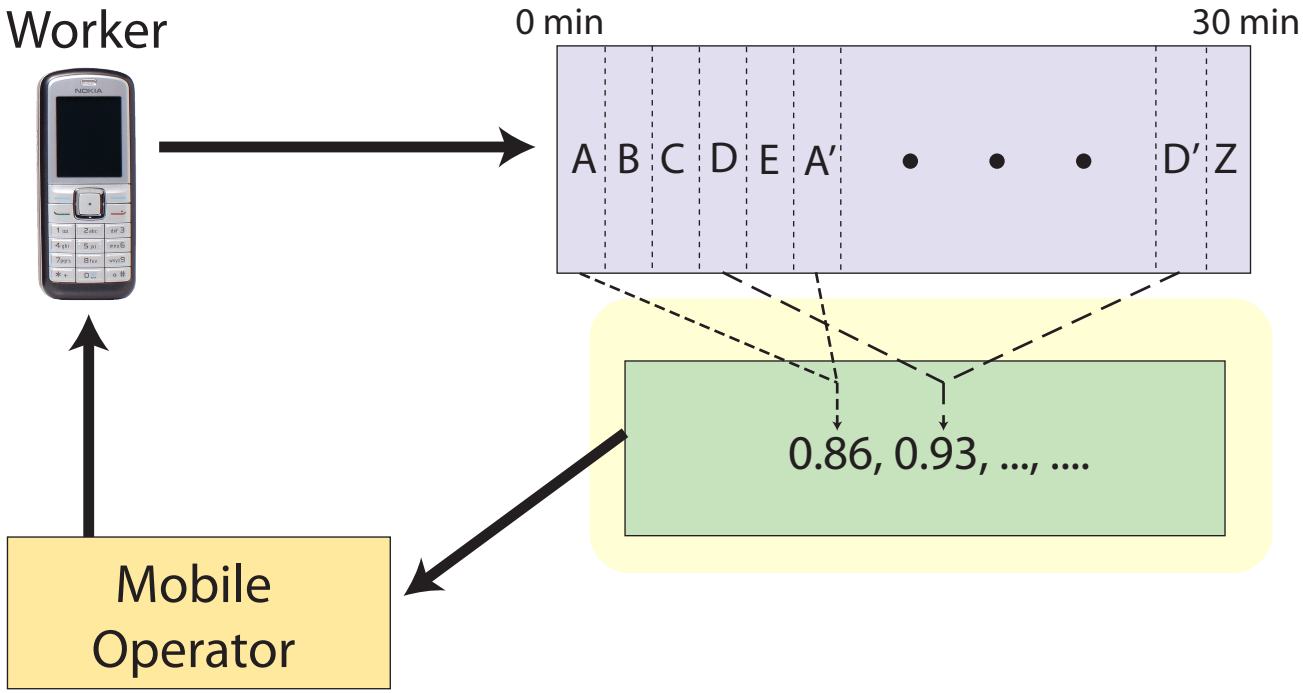


NOKIA
Connecting People

System Overview

Like Mechanical Turk: pay users for validated work (i.e. speech contributions)

1. User flashes CX
   - Gets callback
2. Selects his native language
3. Mimics voice prompts
   - "new transaction"
4. Automatic verification

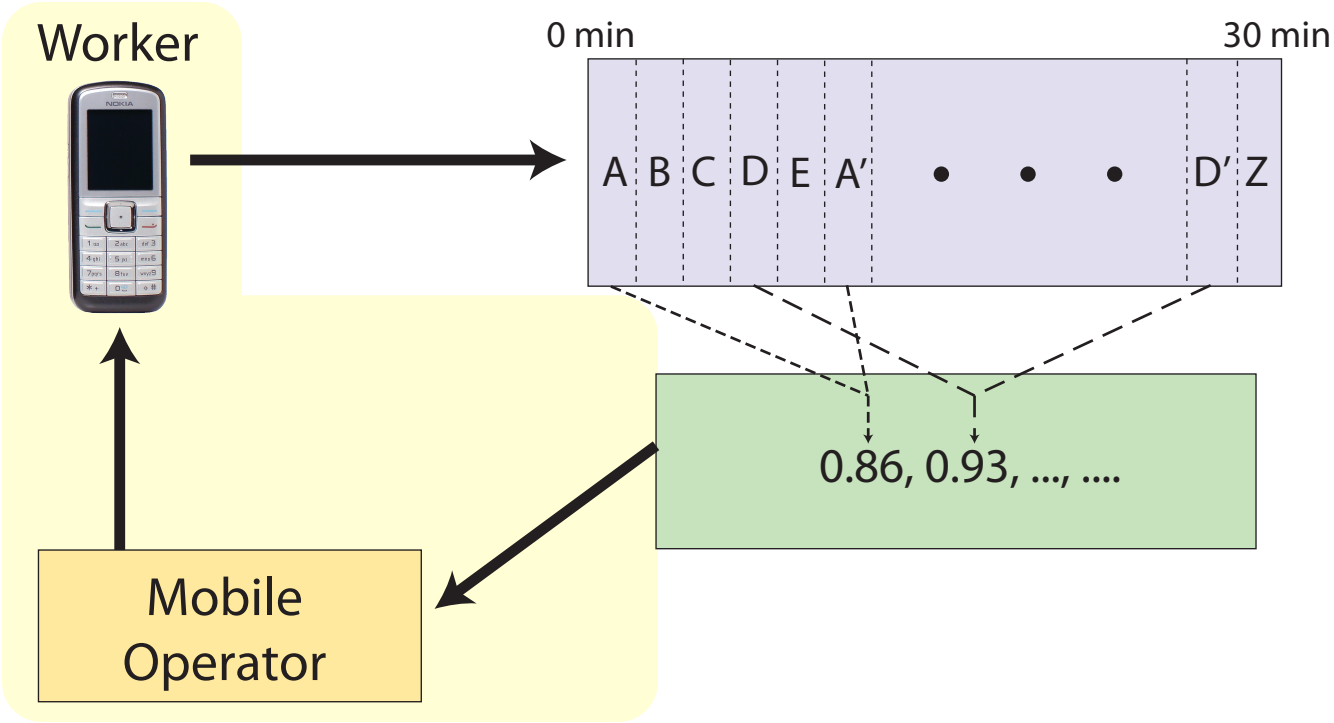


NOKIA
Connecting People

System Overview

Like Mechanical Turk: pay users for validated work (i.e. speech contributions)

1. User flashes CX
   - Gets callback
2. Selects his native language
3. Mimics voice prompts
   - "new transaction"
4. Automatic verification
5. Payment



NOKIA
Connecting People

Automatic Verification

Goal:

- Discard low quality work
- Tolerate noise to improve trust

Previous work (Turk, txteagle, Sarmenta)

- Give k users same task
  - Slow payment

Automatic Verification

Goal:

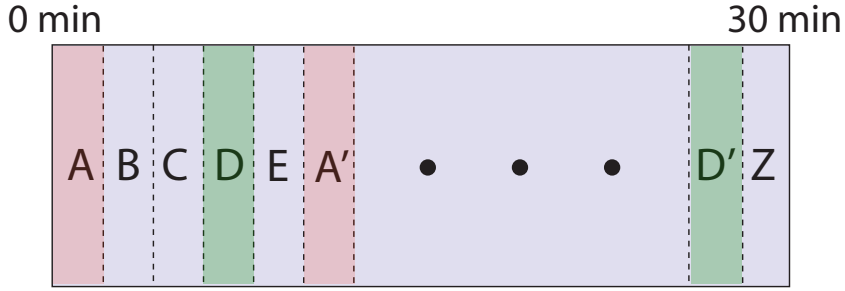
- Discard low quality work
- Tolerate noise to improve trust

Previous work (Turk, txteagle, Sarmenta)

- Give k users same task
  - Slow payment

Our approach: **Intra-session Agreement**

- Make a small fraction of user's queries redundant
- Measure acoustical similarity between each pair
- Examine distribution of similarity scores
  - Like same user saying same word? Accept
  - Else: Reject



`D = s(a,a'), s(d,d'), ..., s(k,k')`

`s(x,y)⇒acoustical similarity`

Automatic Verification

Goal:

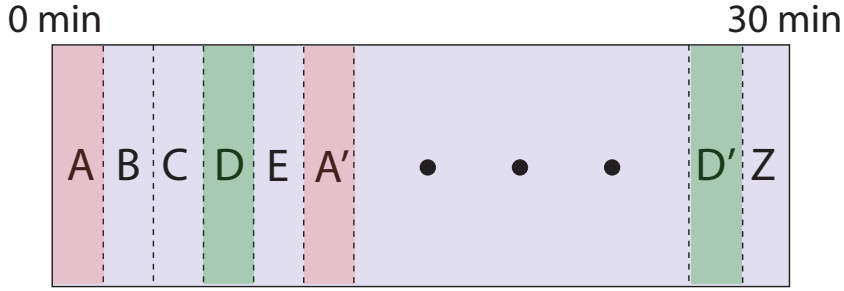
- Discard low quality work
- Tolerate noise to improve trust

Previous work (Turk, txteagle, Sarmenta)

- Give k users same task
  - Slow payment

Our approach: **Intra-session Agreement**

- Make a small fraction of user's queries redundant
- Measure acoustical similarity between each pair
- Examine distribution of similarity scores
  - Like same user saying same word? Accept
  - Else: Reject



`D = s(a,a'), s(d,d'), ..., s(k,k')`

`s(x,y)⇒acoustical similarity`

Can be augmented with other methods

- vs. Gold Standard, vs. Corpus

NOKIA
Connecting People

Prototype Highlights

Prototype Highlights

User Study in Kenya

- 15 users; 55 words x2 (to test auto verification)
- Manually annotated: 1229 valid, 421 invalid

Prototype Highlights

User Study in Kenya

- 15 users; 55 words x2 (to test auto verification)
- Manually annotated: 1229 valid, 421 invalid

vs. Gold Standard

- Too much valid data had low similarity score (high false negatives)

Prototype Highlights

User Study in Kenya

- 15 users; 55 words x2 (to test auto verification)
- Manually annotated: 1229 valid, 421 invalid

vs. Gold Standard

- Too much valid data had low similarity score (high false negatives)

vs. Intra-Session Agreement

- Session valid if 80% utterances valid
- <5% false negative; 25% false positive

Prototype Highlights

User Study in Kenya

- 15 users; 55 words x2 (to test auto verification)
- Manually annotated: 1229 valid, 421 invalid

vs. Gold Standard

- Too much valid data had low similarity score (high false negatives)

vs. Intra-Session Agreement

- Session valid if 80% utterances valid
- <5% false negative; 25% false positive

Take-away

- Intra-Session Agreement plus ...
  (a) immediately vs Corpus
  (b) later with clustering
- Larger validation needed
  - Effect of priming?

Conclusions

Conclusions

Crowd Translator

- Cheaply create phrases-based recognizers for local, low-corpus languages
  - Keep costs low
    - No experts, no lab, existing phones, automated verification
  - User trust
    - Rapid payment for validated work

Conclusions

Crowd Translator

- Cheaply create phrases-based recognizers for local, low-corpus languages
  - Keep costs low
    - No experts, no lab, existing phones, automated verification
  - User trust
    - Rapid payment for validated work

Future Work

- Large-scale corpus in collaboration with Univ. of Nairobi
- New validation algorithms
- Open source: NASI (“with us”) on Sourceforge

Conclusions

Crowd Translator

- Cheaply create phrases-based recognizers for local, low-corpus languages
  - Keep costs low
    - No experts, no lab, existing phones, automated verification
  - User trust
    - Rapid payment for validated work

Future Work

- Large-scale corpus in collaboration with Univ. of Nairobi
- New validation algorithms
- Open source: NASI (“with us”) on Sourceforge

Demo

- +254 711 027 950, +1 617 453 2272

Conclusions

Crowd Translator

- Cheaply create phrases-based recognizers for local, low-corpus languages
  - Keep costs low
    - No experts, no lab, existing phones, automated verification
  - User trust
    - Rapid payment for validated work

Questions?

jonathan.ledlie@nokia.com

Future Work

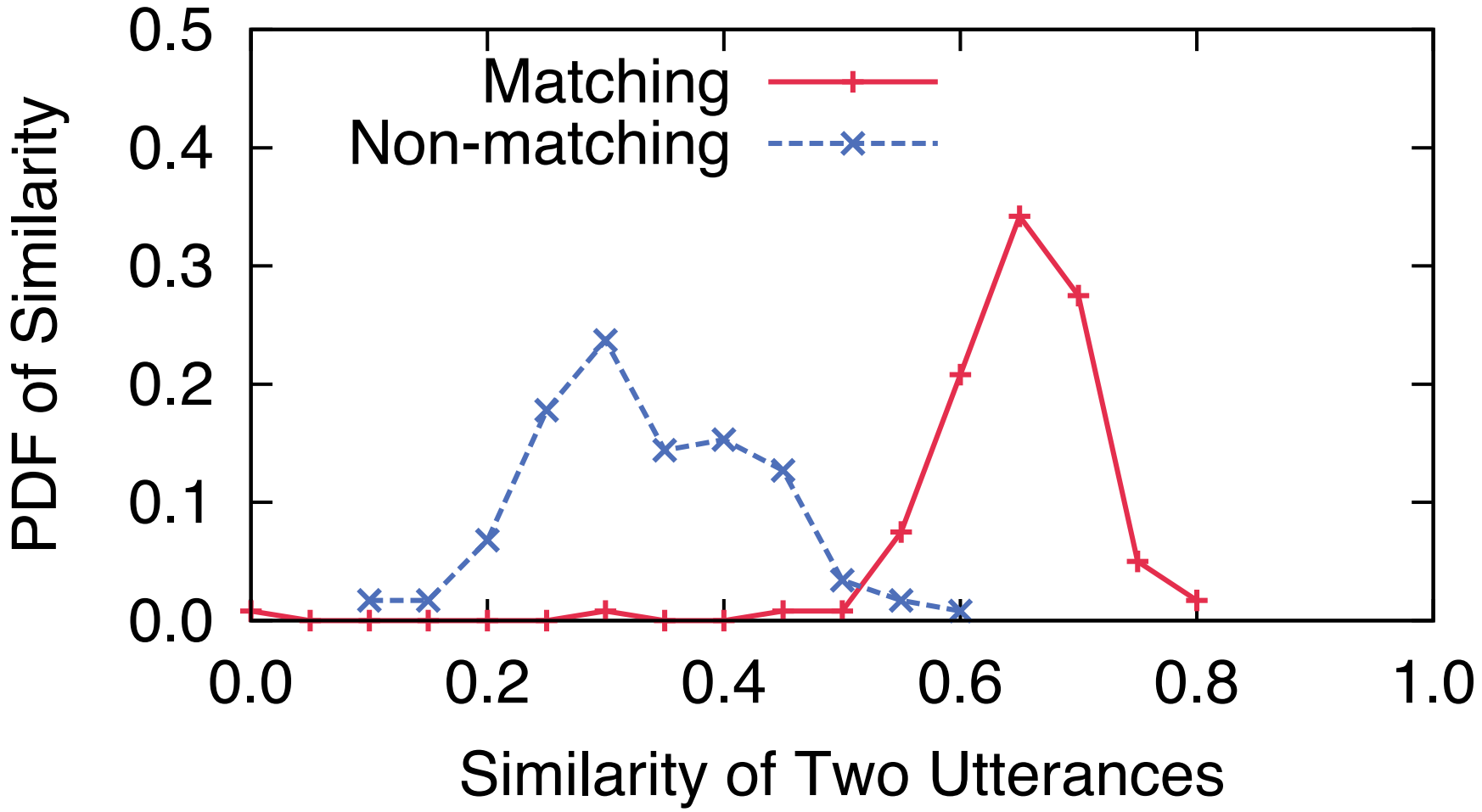
- Large-scale corpus in collaboration with Univ. of Nairobi
- New validation algorithms
- Open source: NASI (“with us”) on Sourceforge

Demo

- +254 711 027 950, +1 617 453 2272

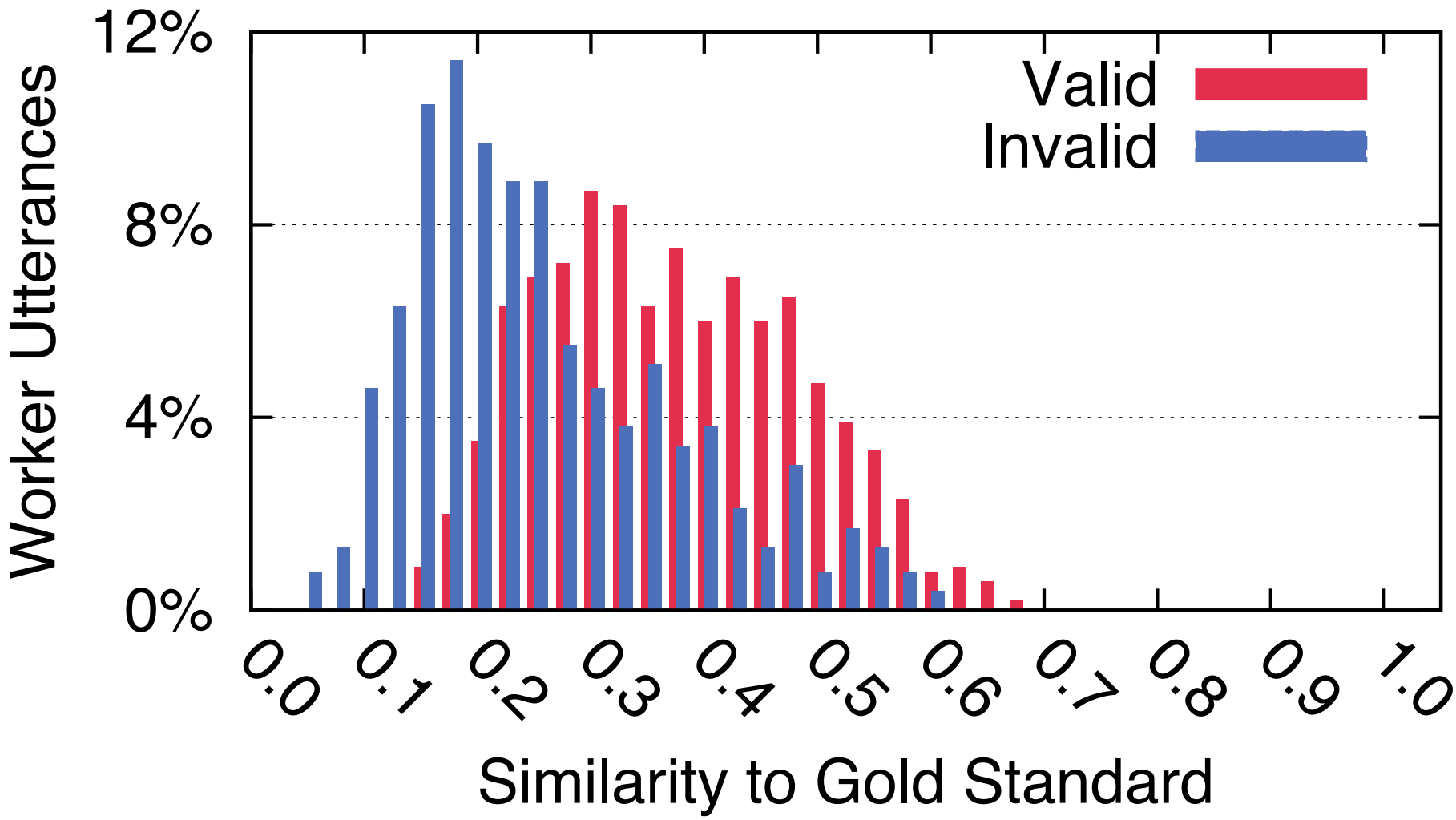
Extra Slides

Similarity Score Distributions



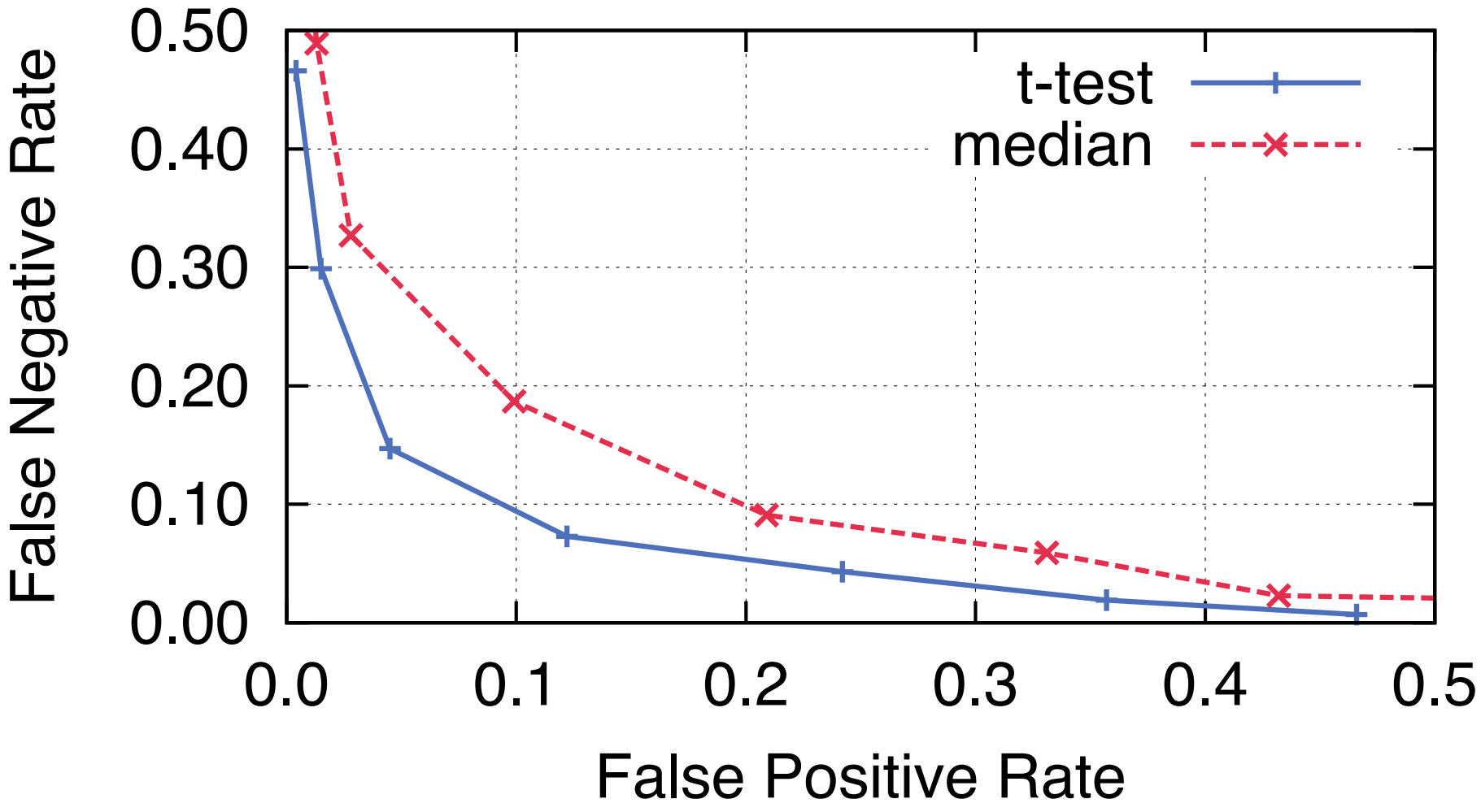
NOKIA
Connecting People

Gold Standard



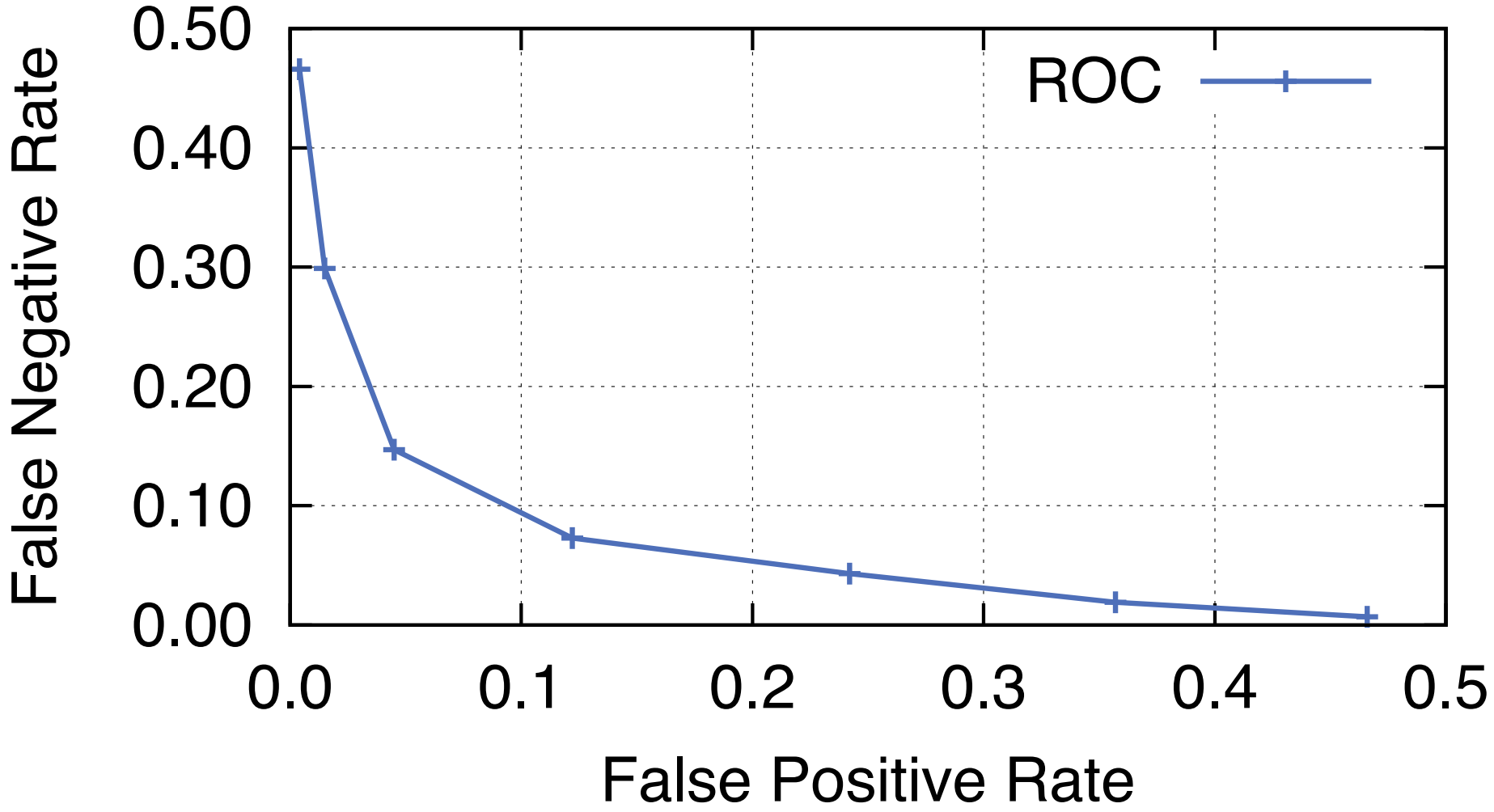
NOKIA
Connecting People

Intra-session Agreement Results



NOKIA
Connecting People

Intra-session Agreement Results



NOKIA
Connecting People

NRC/Cambridge Projects

Three server-side services; prototyping in East Africa (Audio/SMS-based)

- Our focus: User-generated content

Tangaza ("announce" in Swahili)

- Send voice messages to friends, family, and groups
  - e.g., Nairobi taxi drivers, tomato farmers in Uganda
- "Twitter" (social net, status updates) for emerging markets

Crowd Translator

- Apply mechanical turk model to generate input for speech recognizer
  - Micropayments in exchange for small tasks
- Improve device/service localization through speech in many more local languages

Mosoko ("mobile marketplace" in Swahili)

- Post and query advertisements for jobs, apartments, and goods
- "Craigslist" for the Next Billion