ASRU 2007 Panel Discussion

Lattice-Based Keyword Search in Audio

- a bit of intro
- challenges (one industry perspective)

Frank Seide, Peng Yu, Lie Lu, Kit Thambiratnam
Microsoft Research Asia "Audio Information Management and Extraction" Project

indexing uncertainty: lattice-based keyword search

keyword search in audio

lattice-based in audio search

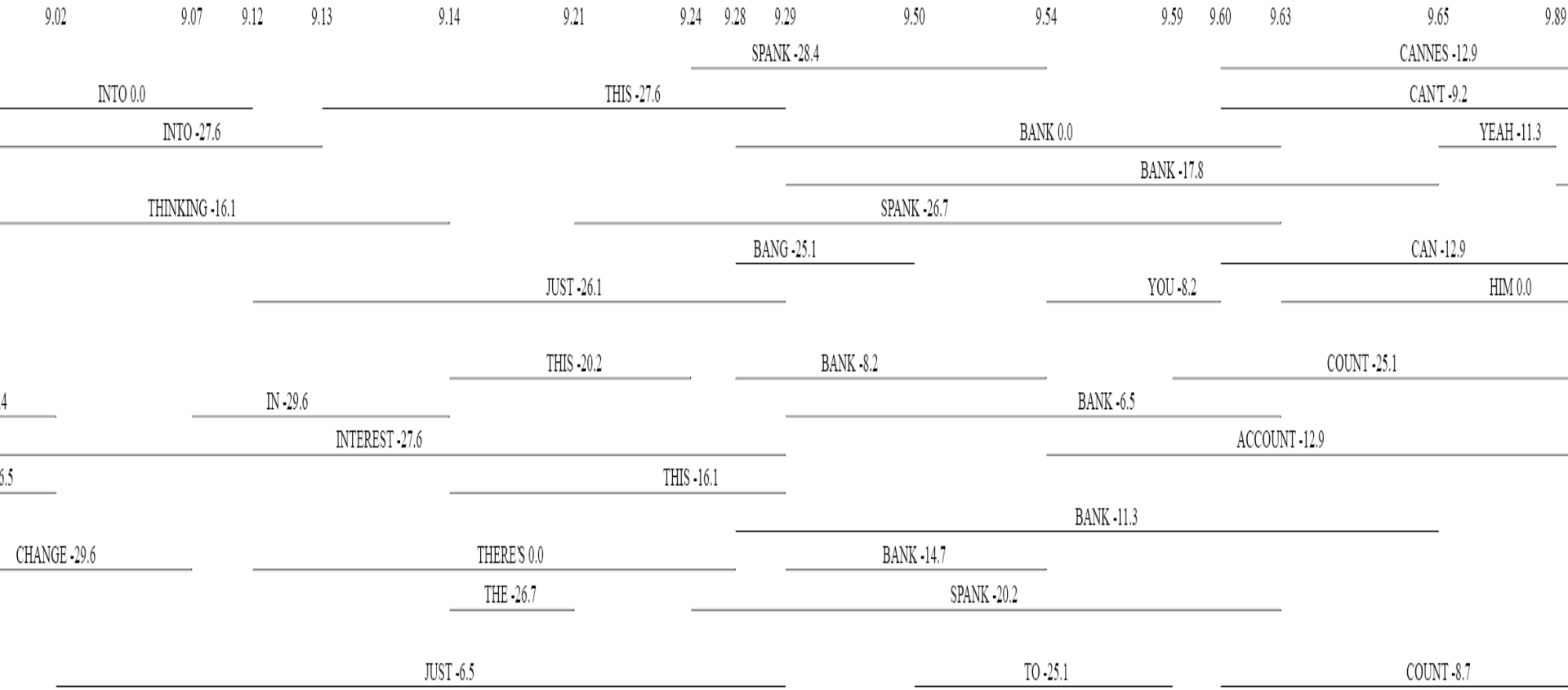
- use of lattices: +36% [Saraclar & Sproat, HLT/NAACL 2004]
- indexing using transducers [Allauzen et al., HLT/NAACL 2004]
- position-specific posterior lattices [Chelba & Acero, ACM 2005]
- word/phonetic hybrid lattices [Yu & Seide, InterSpeech 2004]

lattice indexing benefits

- alternative recognition candidates ➔ recall++
- confidence scores ➔ precision++
- (time information ➔ user experience)

indexing uncertainty: lattice-based keyword search

"into this bank account"



indexing uncertainty: lattice-based keyword search

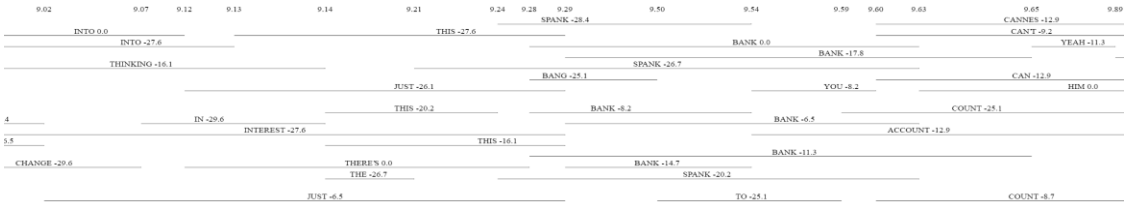
	phrase queries			single-word queries			X AND Y queries			
	FOM	mAP	R75	FOM	mAP	R50	FOM	mAP	R75	index size
STT transcript with confidence	***40.6***	42.7	43.4	36.4	44.2	***45.2***	42.8	26.1	***26.1***	1
raw lattice	***66.1***	67.2	55.7	49.0	55.9	***45.4***	55.6	63.3	***61.6***	1617
TMI	68.4	69.6	58.0	48.7	55.9	45.4	56.1	66.3	63.9	46.2
TMI with pruning	65.7	67.1	58.3	48.2	55.4	45.4	55.0	60.4	61.0	9.9
relative improvement over STT	***62%***	***57%***	***34%***	***32%***	***25%***	***1%***	***30%***	***×2.3***	***×2.3***	**-**

lattice indexing useful for:

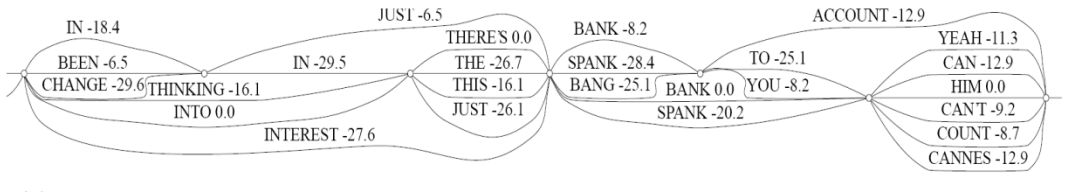
- user-adjustable confidence threshold / FOM metric
- known-item search: high recall
- ad-hoc search: high precision ➔ benefit from AND & phrase queries

indexing uncertainty: lattice-based keyword search

- size reduction: cluster similar word hypotheses (posterior representation)
- e.g. TMI [Yu et al. HLT'06, Seide et al. ASRU'07]
  - allow some boundary inaccuracies: < 1 word (no skip/loop back)
  - group consecutive nodes unless loop is created
  - dynamic programming ➔ minimize #
  - only few extra bits required compared to text indexing



➔

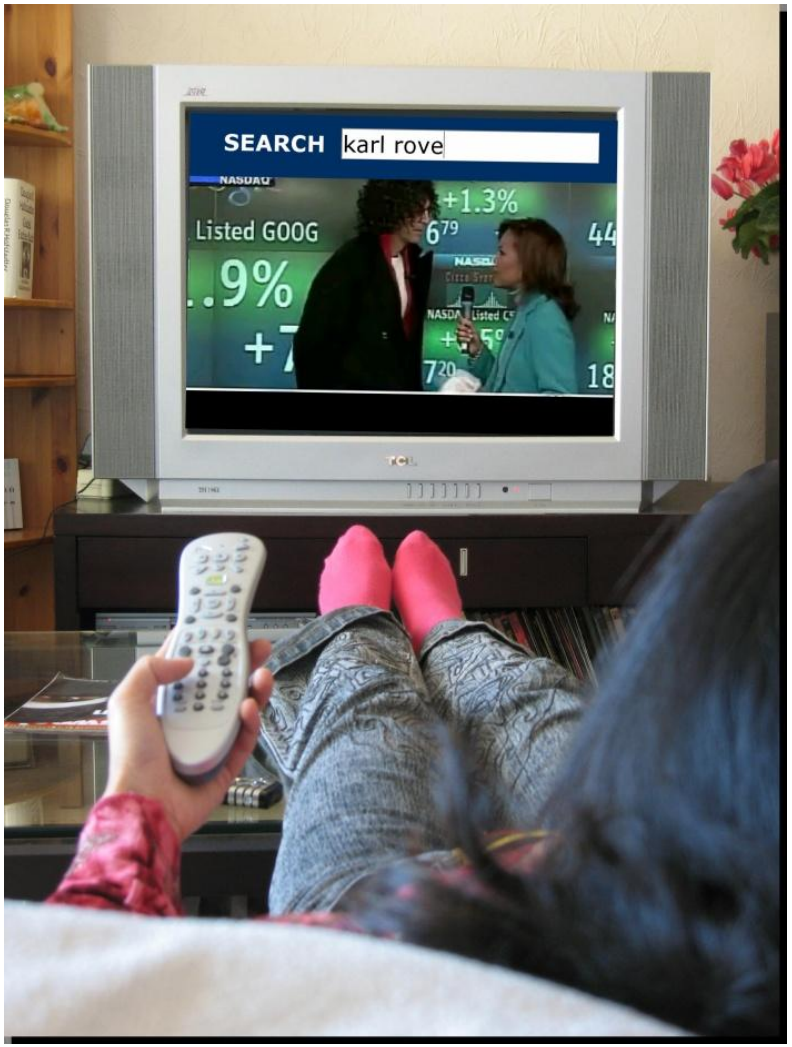


➔ straight-forward to build inverted index (similar to text)

Let’s index the Internet*!*

Microsoft’s video properties on

… and put it where it belongs:
the TV.



killer scenario!

- Audio indexing of Internet video ➔ a killer scenario! …?

27.2%	adult	19.6%	music	8.7%	category	6.9%	celebrities
18.2	s[redacted]	2.2	rihan[n]a	1.8	trailer	1.7	britney [spears]
3.3	p[redacted]	1.7	spice girls	1.4	cat[s]	1.3	avril [lavigne]
0.7	g[redacted]	1.0	beyonce	1.0	funny	1.1	shakira
0.8	girl[s]	0.6	linkin park	0.6	music	0.9	paris hilton
0.5	bikini	0.6	celine dion	0.6	featured	0.5	madonna
2.9%	**TV**	**1.8%**	**news**	**1.5%**	**movies**	**0.4%**	**sports**
0.7	criss angel	0.5	obama	0.6	high school musical [2]	0.2	football
0.4	to catch a predator	0.5	erin burnett	0.3	tweeling	0.2	soccer
0.3	soprano	0.4	dateline	0.2	harry potter		
0.3	heroes	0.2	panda	0.2	predator		
0.2	lost in space	0.2	ron paul	0.2	beowulf		

Source: MSN Video

killer scenario?

- Audio indexing of Internet video ➔ a killer scenario?
  - cost of STT
  - captions; manual transcription relatively cheap; ($90-$200/h)
    commercial content: incentive to make content discoverable (producer-side)
  - mostly entertainment; users don't know what they want; social aspect (recommendation)

- useful scenarios: information focus; low production value;
  text is no alternative; confidentiality
  - podcasts, focus groups, customer communication / feedback
  - lectures, just-in-time learning, internal talks
  - meetings, phone calls, interviews
  - voicemail, audio notes

  ➔ highly varied; ad-hoc recordings; low learning curve; little support

- learning:
  no single one killer app; instead a "long tail" of applications

➔ technology **platform**, audio being **one** feature

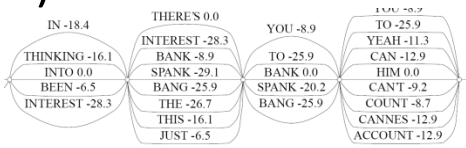
platform challenges

1. easy to deploy, integrate with what's there
2. domain independence
3. suitable for non-technical users

platform challenge #1: integration with what's there

- desire to re-use investments in text indexers
- lattices cannot be indexed with text indexers (no word-position concept)

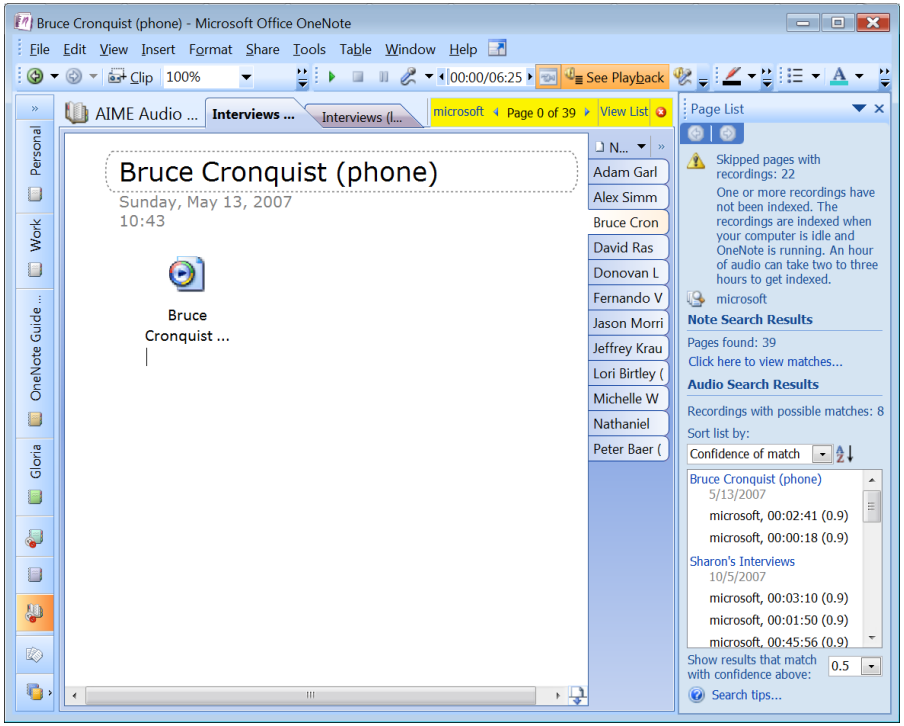
- sausages: infeasible due to ε edges
- solution: [Seide et al, ASRU'07]
  - align and bin *n*-grams to *n* consecutive positions (TALE)
  - no code change inside required
  - only pre/postprocessing



- learning: solve organizational problem with technology…

platform challenge #2: domain independence

- OOV ➔ phonetic approach:
  - phonetic lattices / phonetic search [Seide et el., HLT/NAACL'05]
  - index phonotactically allowable 5 grams, collapse into segments, use as fast match
  - shipping today in Microsoft Office OneNote 2007



platform challenge #2: domain independence

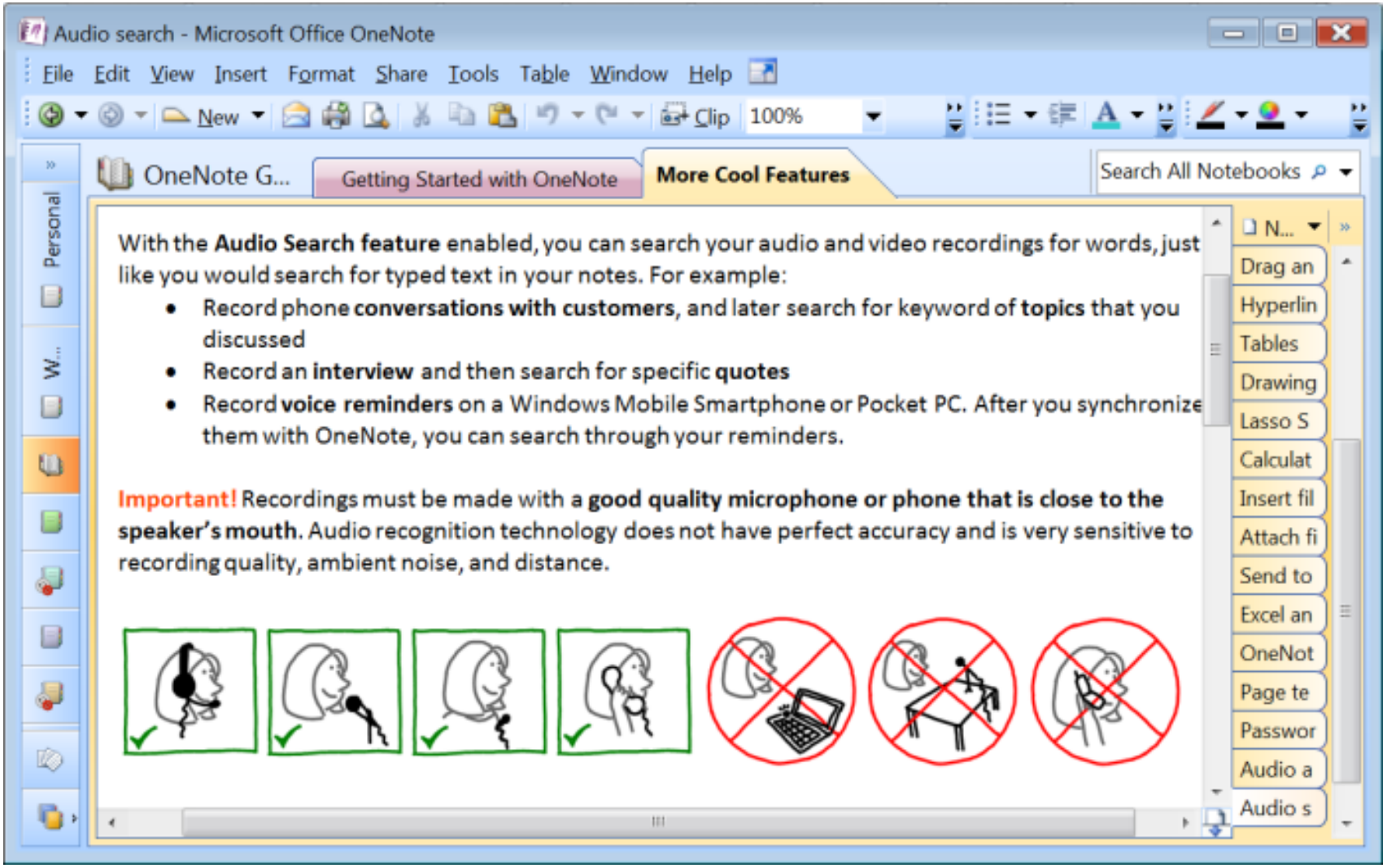
- OOV ➔ phonetic approach:
  - phonetic lattices / phonetic search [Seide et el., HLT/NAACL'05]
  - index phonotactically allowable 5 grams, collapse into segments, use as fast match
  - shipping today in Microsoft Office OneNote 2007

setup (iCampus)	FOM:	1.6 h	→	16 h	→	160 hours	
phonetic search		76%	→	68%	→	57%	-9% points / 10 x
word lattice search		67%	→	65%	→	60%	-4-5% points / 10 x
hybrid		84%	→	78%	→	69%	

- scales poorly in size and precision

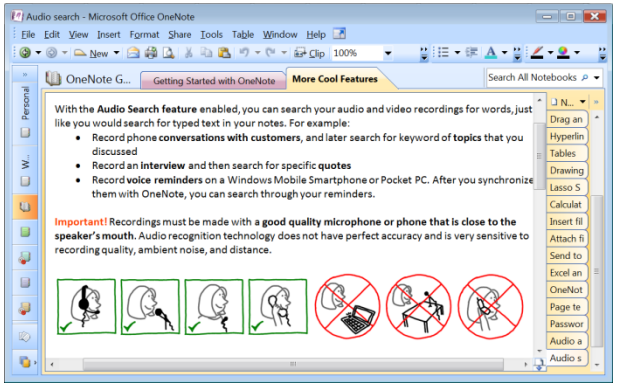
➔ remaining ASR-related challenge:
OOV lattice indexing that scales w.r.t. size *and* precision

platform challenge #3:
“do it yourself” recognition



platform challenge #3: “do it yourself” recognition

- mainstream: audio indexing → as easy as text indexing



- “do it yourself” recognition
  - vertical “long tail” apps
  - no research team building the app for you

- specific challenges:
  - ad-hoc recording conditions → distant talking (reverberation, noise)
  - vocabulary / language model verticalization → use context (keywords, docs, e-mails…)
  - capitalize on user audio → unsupervised adaptation/learning

→ “do it yourself” recognition: conversational ASR in ad-hoc recording conditions for non-technical users

key takeaways

- indexing keywords? → use lattices
  - lattices significantly improves accuracy
  - indexing word lattices no major challenge anymore
- the problem: killer scenario?
  → platform / one feature
  - enable “long tail” of customers to build audio-search apps
  - integration with existing infrastructure, text apps, etc.
- what is needed to succeed:
  - large-scale phonetic / vocabulary-independent indexing
  - “do it yourself recognition”

  ➔ *a “killer” research program!*