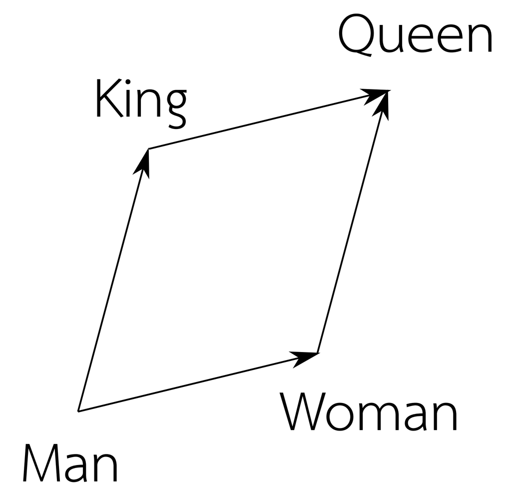
Discovering atomic, compositional, propositional language model by unsupervised learning

Masataro Asai (IBM)
Peter Tang(IBM)

Continuous distributed word embedding

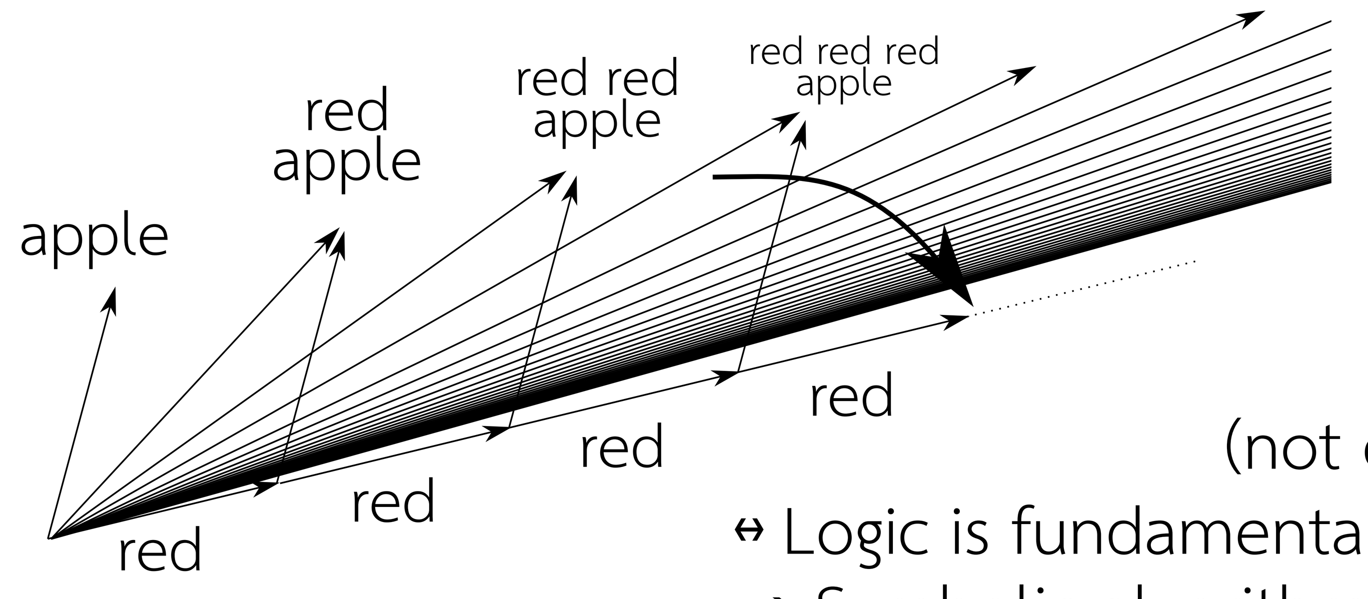


Nice vector-space property:
e.g., king + (woman – man) = queen
(despite critisisms [Nissim et al, 2020])

Logical correctness → ?
e.g. RNNs emit redundant entries:
"I want an apple apple apple …"
e.g. difficulty capturing the negation

It could be confused by imbalance/repetition

$$\cos\angle(v(\text{red}),\ v(\text{red})\cdot n+v(\text{apple})) \to 0 \quad (n\to\infty)$$



(not confused by it)
⇔ Logic is fundamentally discrete
→ Symbolic algorithms
= Discrete math (e.g. graph search)

Instead, we want a better vector operation (+) that satisfy:
v(red)+ … +v(red)+v(apple) = v(red)+v(apple).
→ The (+) operation uses the set operations, ∪, ∩, /.

But ... symbolic NLP?

Structural pattern matching with human symbols

Document → Parser (e.g. AMR) → Prolog-like query

```
(w / want-01
   :arg0 (b / boy)
   :arg1 (a / apple))
```

```
(:- (above ?x ?y)
    (or (on ?x ?y)
        (exists (?z)
          (and (on ?z ?x)
               (above ?y ?z)))))
```

Human symbols are brittle = lacks the vector space property
E.g. Symbol "king" is opaque
(no internal structure telling its gender)

Wittgenstein's "Logical Atomism" (blue book)

There must be a set of atomic concept (e.g. gender)
= **propositional symbols** in human language
→ Learned propositional symbols address the weaknesses of both paradigms (neural / symbolic)
→ Discrete word embedding (from continuous relaxation)

Continuous relaxation of discrete activation: BinaryConcrete VAE (Maddison et al ICLR17)

$$\text{BinConcrete}(x) = \text{Sigmoid}(\ (x + \log u - \log(1-u))\ ) / \ \tau\ )$$
$$u \sim \text{Uniform}(0,1)$$
$$\tau \downarrow 0$$
$$\text{BinConrete}(x) \to \text{Heaviside}(x)$$

Discrete Vector arithmetic

Discretizing the word vector alone is insufficient.
Our discrete embedding has 2 vectors: **Add**, **Del**
Add represents the facts that are made True by the word
→ corresponding to increasing the value of a property
Del represents the facts that are made False by the word
→ corresponding to decreasing the value of a property

If a state vector is [gender, social_status],

add(man)=[0, 0],	del(man)=[1, 0],
add(woman)=[1, 0],	del(woman)=[0, 0],
add(king)=[0, 1],	del(king)=[1, 0],
add(queen)=[1, 1],	del(queen)=[0, 0],

Adding discrete vectors = executing the STRIPS actions sequentially.

STRIPS classical planning (Fikes 71): state transition defined as:
Let P a set of propositional variables (facts). P={have_$200k, have_ferrari, happy}
A **state** s is a subset of P : s ⊆ P , s={have_$200k}
where each proposition p ∈ s is the one that holds (is true) in s.
An **action** a is a tuple <pre(a), add(a), del(a)>:
→ preconditions, add effects, delete effects. (we don't use preconditions)
Given the current state s and an action a, the next state t is:
t = apply(a, s) = (s / del(a))∪ add(a)
del(buy_ferrari) = { have_$200k },
add(buy_ferrari) = { have_ferrari, happy }.

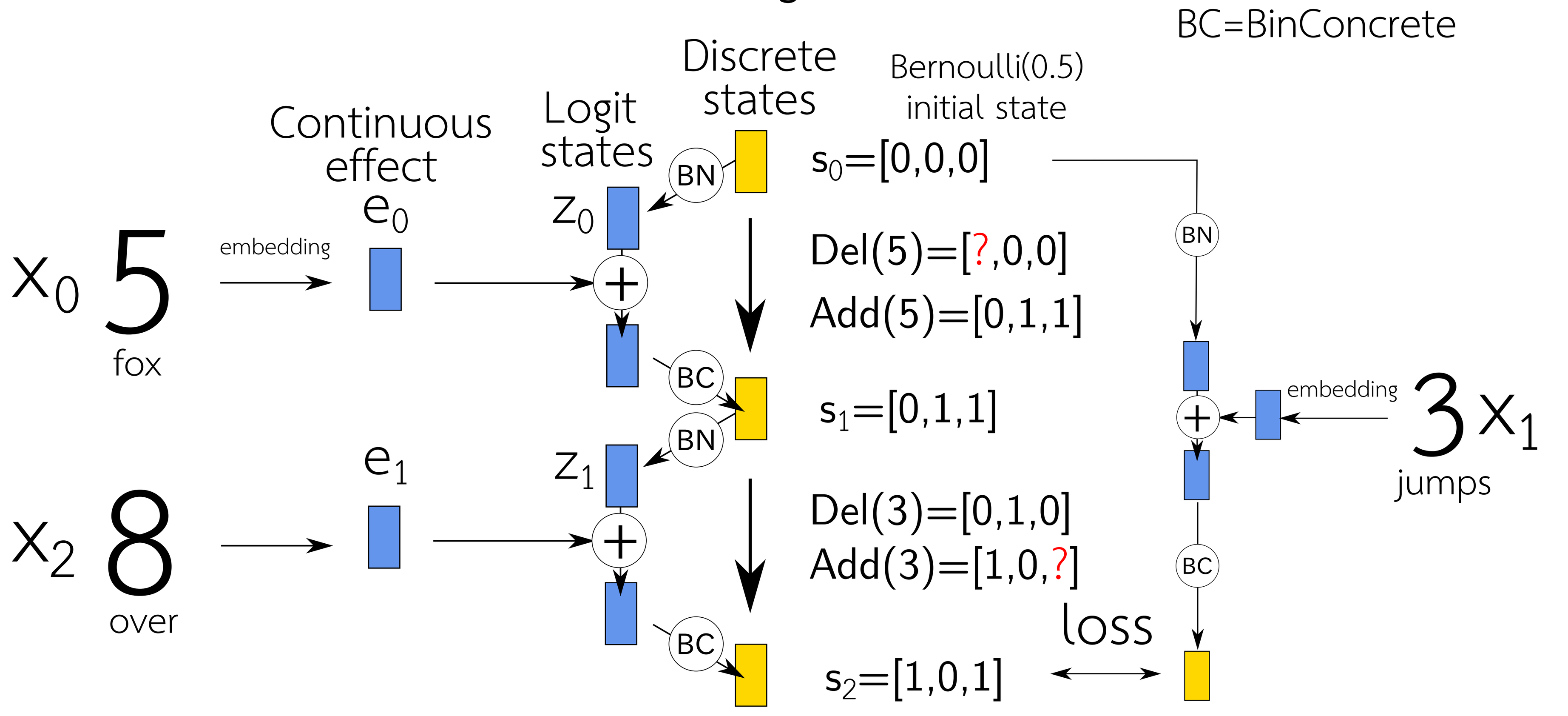
v(man) + v(king):
t = s / del(man) ∪ add(man) / del(king) ∪ add(king)
= s / {gender} ∪ {social_status} ←*The effects do not accumulate*

v(man) - v(king) by **undoing** "king"
t = s / del(man) ∪ add(man) / add(king) ∪ del(king)
= s / {gender} ∪ {} / {social_status} ∪ {gender}
= s / {social_status}

v(king) + v(woman) – v(man) = v(queen)
t = s / {gender} ∪ {social_status}
/ {} ∪ {gender} / {} ∪ {gender} = s ∪ {gender, social_status}
add(queen)=[1, 1]

Discrete state CBOW (3-gram)

BN=BatchNorm
BC=BinConcrete



k-NN (excluding itself) (prelim. results)

car	:	cheapest	subcompact	saloon	mechanic
motorcycle	:	motorbike	lawnmower	50cc	megane
bicycle	:	bike	stationary	bicycles	bikes
lamborghini	:	automobili	murcielago	tonino	gallardo
ferrari	:	massa	rosso	gtb	scuderia
maserati	:	quattroporte	8688	mobtown	twinn
fiat	:	punto	abarth	cinquecento	eni
renault	:	megane	clio	hyundai	nissan
bmw	:	sauber	z4	oracle	daimler
mercedes	:	benz	benzes	bmw	cosworth
audi	:	rs4	acura	a6	a4
toyota	:	nissan	prius	lexus	aygo
honda	:	hyundai	motor	toyota	fireblade
mazda	:	cx	camaro	hatchback	changan
nissan	:	altima	micra	qashqai	xterra
subaru	:	wrx	isuzu	impreza	volkswagen
ford	:	mondeo	madox	motor	chadds
gm	:	chrysler	digiovanni	motors	delphi
chevrolet	:	gmc	equinox	hhr	sedan
suzuki	:	ichiro	vitara	maruti	tas
kawasaki	:	kisen	boonsboro	phrasemaking	takfiris
ducati	:	pramac	meddwl	200c	icex
yamaha	:	rider	aprilia	suzuki	honda

king	:	tut	birendra	gyanendra	lich
queen	:	sirikit	latifah	rania	ifrica
prince	:	ashwell	vajiralongkorn	dipendra	epiphanny
princess	:	diana	leia	galyani	cruises
sea	:	salton	okhotsk	sargasso	urchin
lake	:	wobegon	okeechobee	titicaca	havasu
river	:	zambezi	yangtze	mekong	yalu
pond	:	rattlesnake	tailings	scum	blacktop
island	:	brownsea	rhode	basilan	necker
mountain	:	yucca	brokeback	sobek	cordillera
hill	:	menwith	muswell	notting	ch2m
valley	:	saucon	willamette	hybla	rift
forest	:	mangrove	tongass	inyo	petrified
woods	:	tiger	bogeyed	watney	stricker
apple	:	iwork	iphone	quicktime	itunes
grape	:	varieties	pomegranate	vines	oysters
orange	:	county	bright	magenta	monterey
potato	:	potatoes	gnocchi	crispy	tart
carrot	:	parfait	cornbread	tarragon	parsnips
onion	:	zucchini	pickled	broth	tacos
garlic	:	thyme	potatoes	cloves	chopped

training data: 1-billion monolingual dataset

Future work: logical reasoning in the embedded space using symbolic methods

Shortest path finding / planning

Q. What is the shortest phrase for an "black adult male bovine"?
A. black ox

Negation via boolean-SAT

background: (not red) apple.
Q: X apple.
A: X=green

Future work: extension to RNN, attention model (e.g. BERT)