smart reply and implicit semantics

Matthew Henderson and Brian Strope
Google AI

collaborators include:

Rami Al-Rfou, Yun-hsuan Sung

Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar

Balint Miklos, Ray Kurzweil

… and many others

Machine learning works when it **generalizes** to things unseen in training.

training:

```
How do you commute to work? -> I ride my bike.
What's your favorite color? -> I like red.
```

testing:

```
Do you like red bikes? ->
```

generalization strategies

explicit semantics: discrete frames, slots and values

implicit semantics: continuous vectors

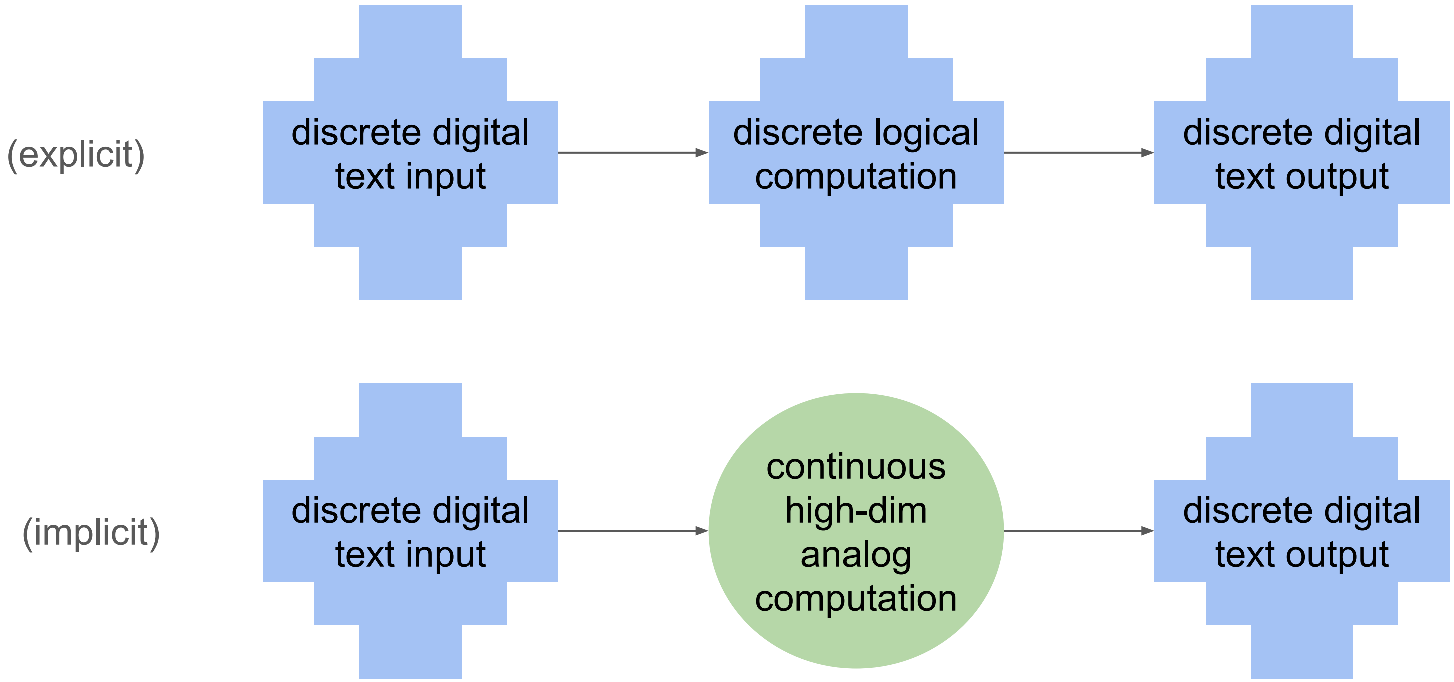
explicit semantics

specified by humans (often for a task)
debuggable
fundamental to understanding (?)

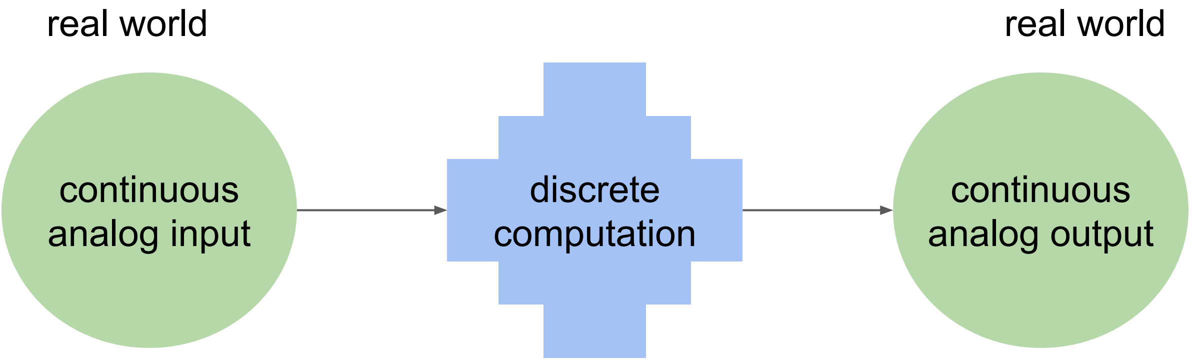
implicit semantics

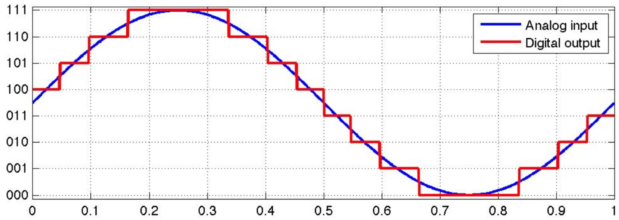
not specified
derived during training
emergent
natural efficiency for compression and generalization (?)

explicit and implicit semantics: analog / digital

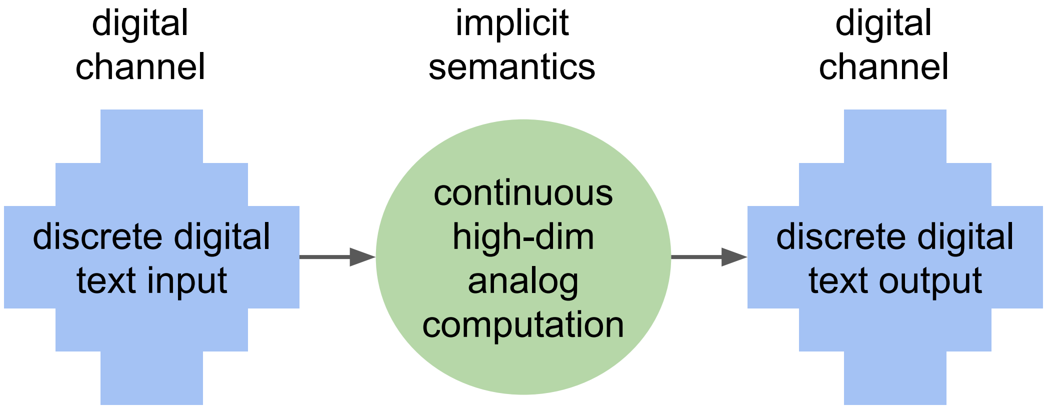


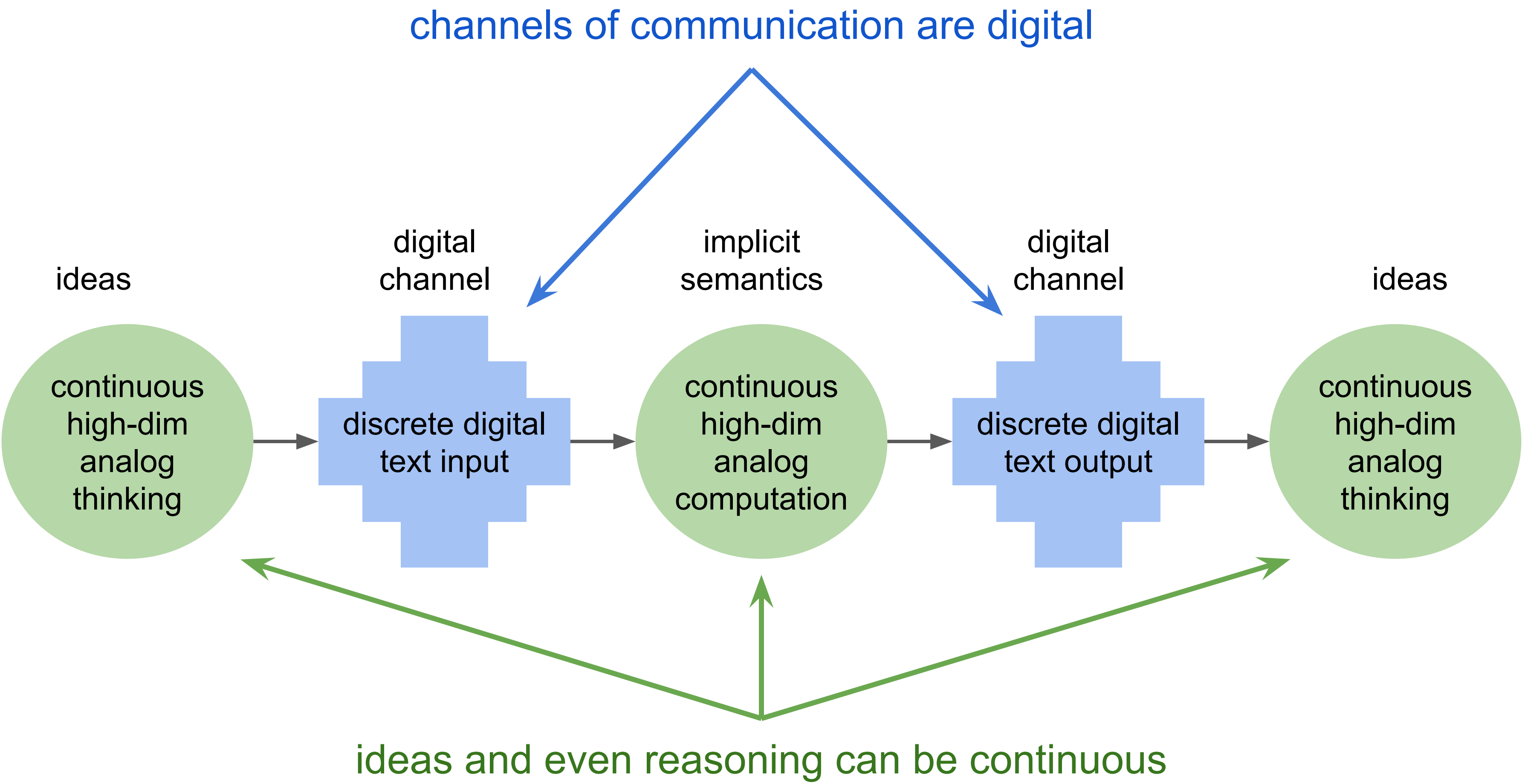
signal processing (opposite)





implicit semantics -- why go back to continuous?




channels of communication are digital
ideas
digital
channel
implicit
semantics
digital
channel
ideas
continuous
high-dim
analog
thinking
discrete digital
text input
continuous
high-dim
analog
computation
discrete digital
text output
continuous
high-dim
analog
thinking
ideas and even reasoning can be continuous

training task with “semantic pressure”

next sentence prediction, reply prediction

I saw a really good band last night.

→

training task with “semantic pressure”

next sentence prediction, reply prediction

I saw a really good band last night.

→

They played upbeat dance music.

training task with “semantic pressure”

next sentence prediction, reply prediction

I saw a really good band last night. →

It often rains in the winter.
On Thursdays we like to go out.
They played upbeat dance music.
The tree looks good to me.
Did you get a new car?
My son likes to windsurf.
Looking forward to lunch.

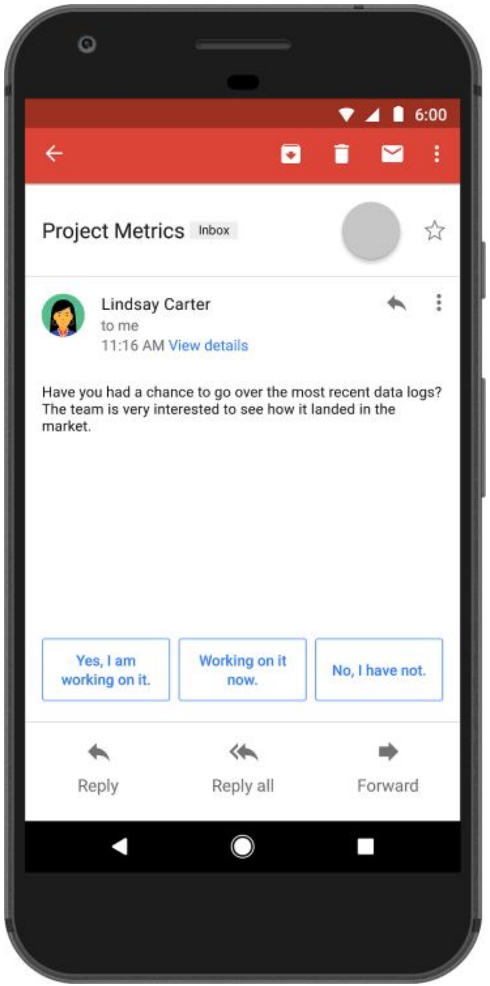
an initial application: smart reply

Smart reply for Inbox & Gmail

feature that suggests short responses to emails

initial system used an LSTM to read input email, and did a beam search over the whitelist

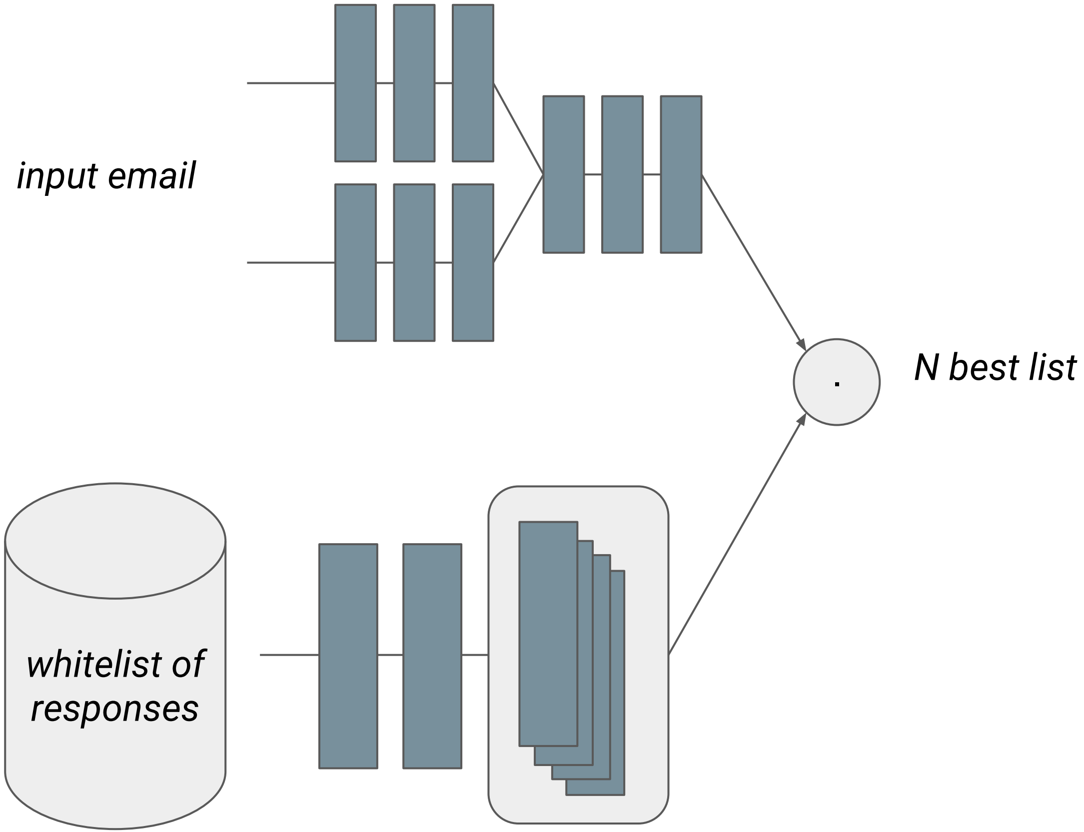
measure 'suggest conversion', %age of times shown suggestions are clicked



The direct smartreply system

trained to give a high score for the response found in the data, low score for random responses

final score of an email and response is a dot-product of two vectors



INNOVATION + ASSISTANCE

Training a dot-product model

network encodes a batch of input emails to vectors:

$$x_1 \quad x_2 \quad \dots \quad x_N$$

and responses to vectors:

$$y_1 \quad y_2 \quad \dots \quad y_N$$

$x_1.y_1$	$x_1.y_2$	$x_1.y_3$	$x_1.y_4$	$x_1.y_5$
$x_2.y_1$	$x_2.y_2$	$x_2.y_3$	$x_2.y_4$	$x_2.y_5$
$x_3.y_1$	$x_3.y_2$	$x_3.y_3$	$x_3.y_4$	$x_3.y_5$
$x_4.y_1$	$x_4.y_2$	$x_4.y_3$	$x_4.y_4$	$x_4.y_5$
$x_5.y_1$	$x_5.y_2$	$x_5.y_3$	$x_5.y_4$	$x_5.y_5$

Training a dot-product model

the N x N matrix of all scores is a fast matrix product.

10% absolute improvement in 1 of 100 ranking accuracy over binary classification.

$x_1.y_1$	$x_1.y_2$	$x_1.y_3$	$x_1.y_4$	$x_1.y_5$
$x_2.y_1$	$x_2.y_2$	$x_2.y_3$	$x_2.y_4$	$x_2.y_5$
$x_3.y_1$	$x_3.y_2$	$x_3.y_3$	$x_3.y_4$	$x_3.y_5$
$x_4.y_1$	$x_4.y_2$	$x_4.y_3$	$x_4.y_4$	$x_4.y_5$
$x_5.y_1$	$x_5.y_2$	$x_5.y_3$	$x_5.y_4$	$x_5.y_5$

$$\boldsymbol{x}_i = DNN(\ n\text{-grams of email } i\ )$$

$$\boldsymbol{y}_i = DNN(\ n\text{-grams of response } i\ )$$

$$S_{ij} = \boldsymbol{x}_i \cdot \boldsymbol{y}_j$$

$$P(\text{ response } j \mid \text{ email } i\ ) \propto e^{Sij}$$

$$-\ log\ P(\text{ example } i\ ) = -\ S_{ii} + log\ \Sigma_j\ e^{Sij}$$

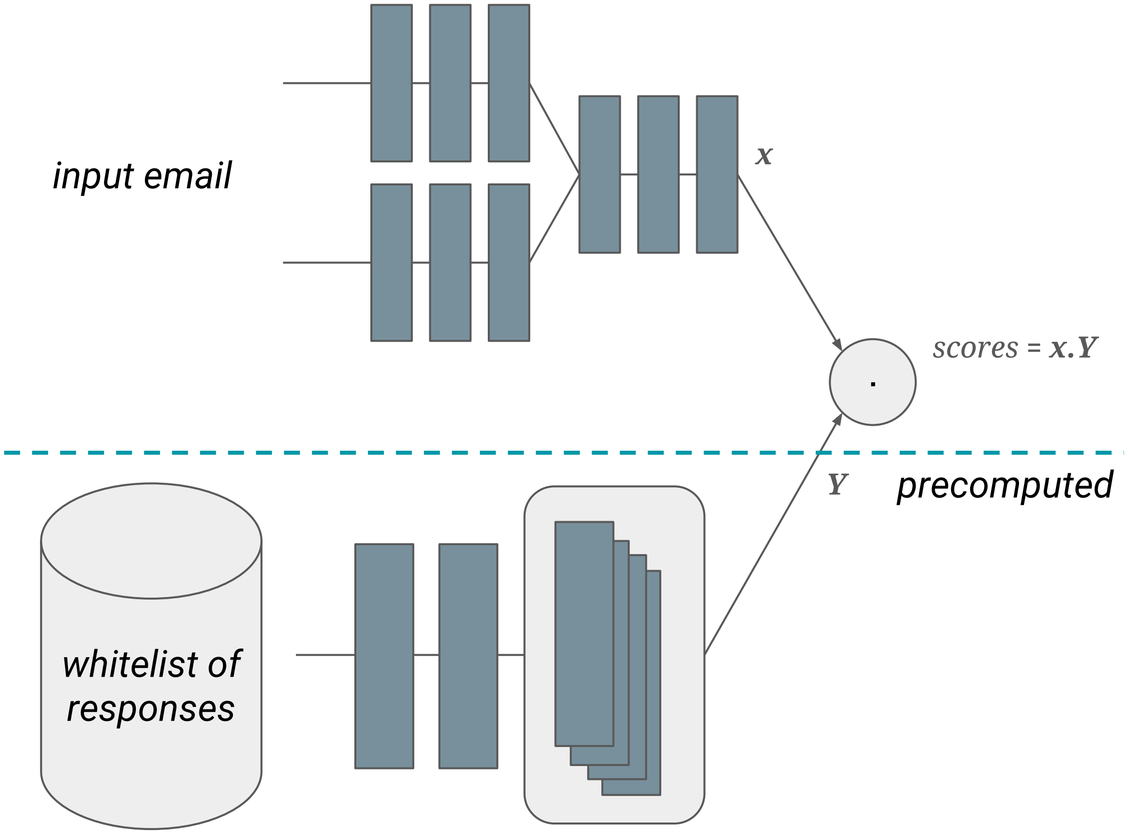
"dot product loss"

INNOVATION + ASSISTANCE

Precomputation for dot product model

the representations of the whitelist $\boldsymbol{Y}$ can be precomputed

approximate nearest neighbor search can speed up the top N search



INNOVATION + ASSISTANCE

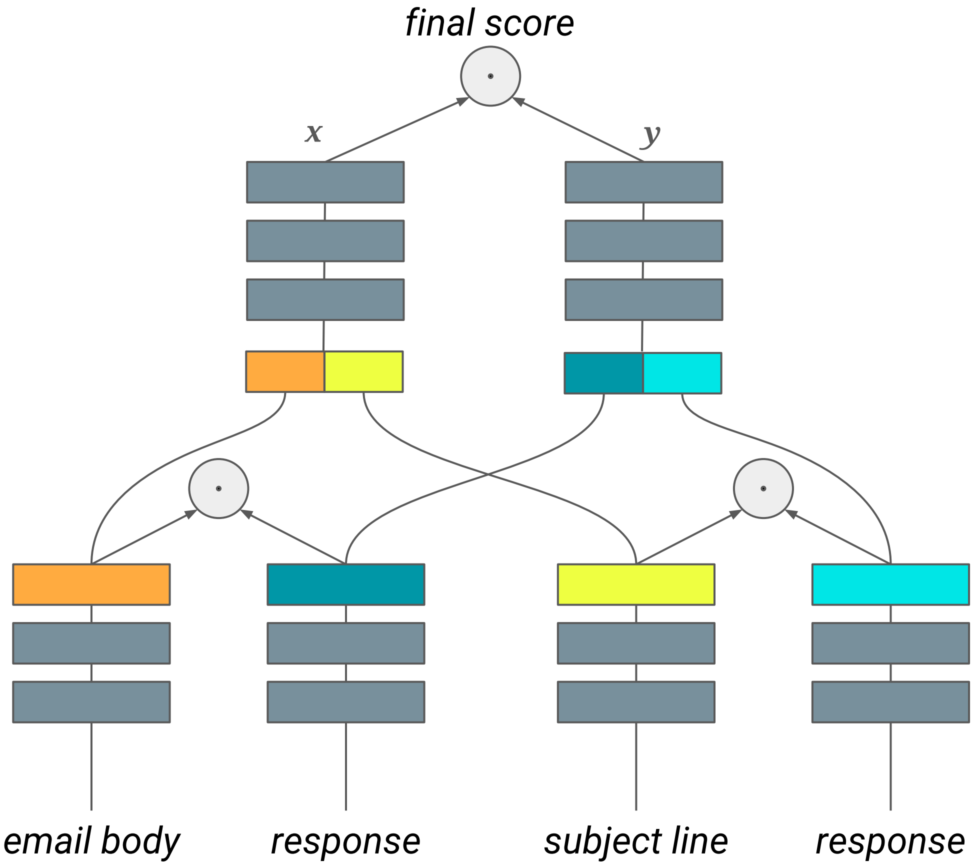
Multi-loss dot product model

each feature predicts the response on its own, then are combined

originally used to inspect importance of each feature

gives extra depth and hierarchy

10% absolute improvement in 1 of 100 ranking accuracy over concatenating input features and using a single loss



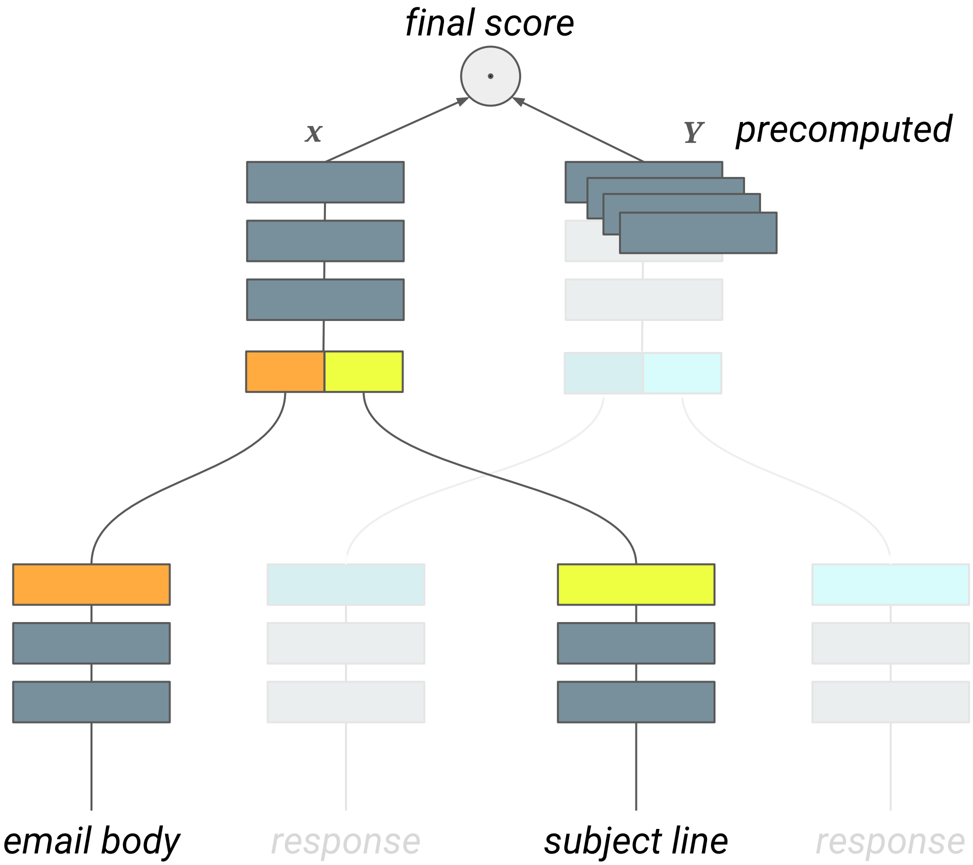
INNOVATION + ASSISTANCE

Multi-loss dot product model

each feature predicts the response on its own, then are combined

originally used to inspect importance of each feature but gives extra depth and hierarchy

10% absolute improvement in 1 of 100 ranking accuracy over concatenating input features and using a single loss



Latency

LSTM	DNN	Dot product + DNN	Dot product only	Approximate search
	5x latency	0.1x	0.02x	0.01x
Beam search over prefix trie of whitelist.	Score everything on the whitelist with a fully-connected DNN.	Use dot product model as first pass to select 100, then score with DNN.	Use improved *multi-loss* dot product model in one pass of scoring.	Speed up top N search in dot product space using an efficient nearest neighbor search.

(non-LSTM systems can achieve *suggest conversion* around 4% higher than LSTM)

Response biases

initial “direct” system got about half the number of clicks of LSTM baseline

language model bias improves clicks

probability-of-click model on actual smartreply emails helps more

combinations improve click rate above LSTM baseline

“ *Thank you so much for the wonderful gifts.* ”

Glad you liked the gifts.

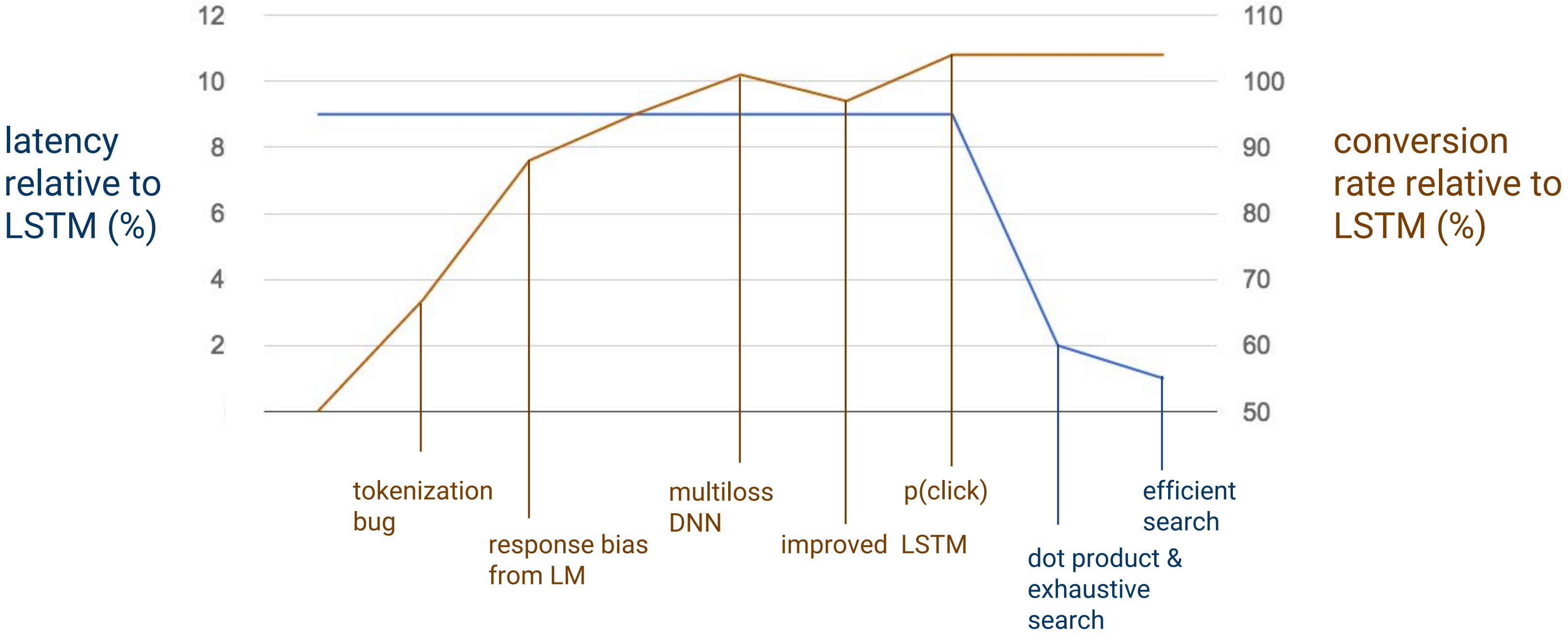
Our pleasure!

You are very welcome!

You're welcome!

Thank you!

Quality and latency progress



8 weeks

conclusions

“implicitly semantic” representations are useful

beam search isn’t always necessary (simple works too)

having user quality signals (like clicks) can be very helpful

Thank you!