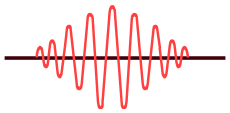
Recurrent Neural Nets Applications

Thang Vu

Automatic Speech Recognition

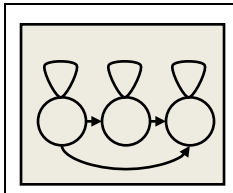
Search
how to efficiently try all W

$$\arg\max_W P(W \mid X) = \arg\max_W \frac{P(W)p(X \mid W)}{P(X)}$$



Signal Processing

Input Speech



I /i/
you /j/ /u/
we /v/ /e/

I am
You are
She are

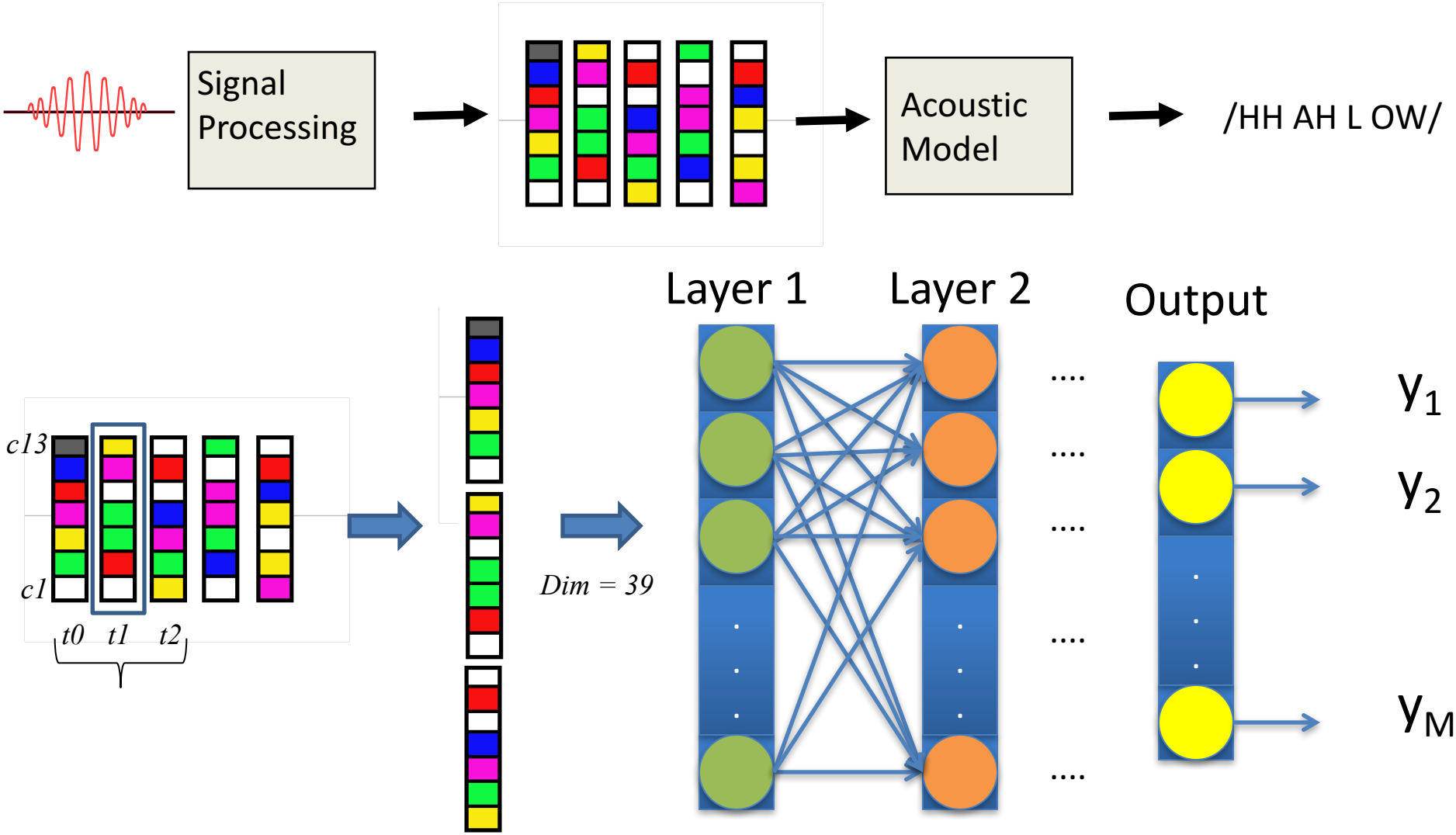
Output Text

Hello world

Acoustic Model +
Pronunciation Dict

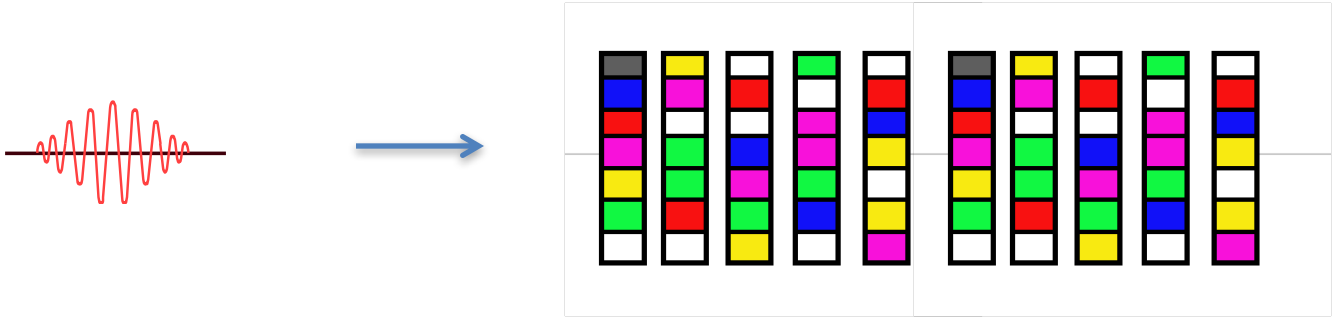
Language Model

Deep Neural Nets for Acoustic Models



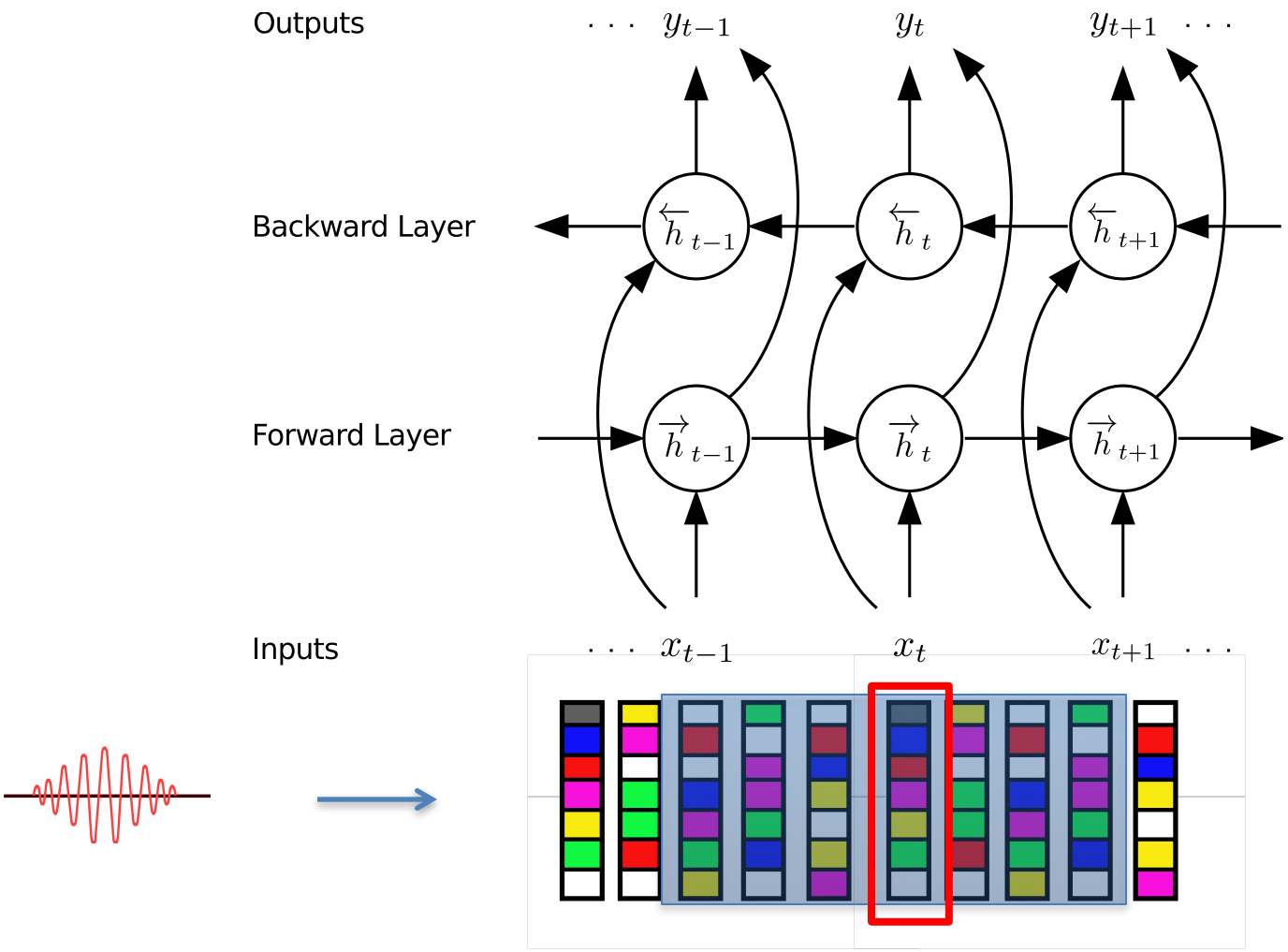
RNN for Acoustic Modelling

- RNN can be used in speech recognition
- Speech signal → Sequence of features



- RNN has shown to improve the ASR performance
  - Graves et al, 2013
  - Now, it belongs to one of the state-of-the-art techniques

RNN for Acoustic Modelling



Google Home Acoustic Modelling

- Google Home uses RNN-LSTM, 2017
- Trained with 18k hours of speech data
- 4 LSTM layers with 1024 cells

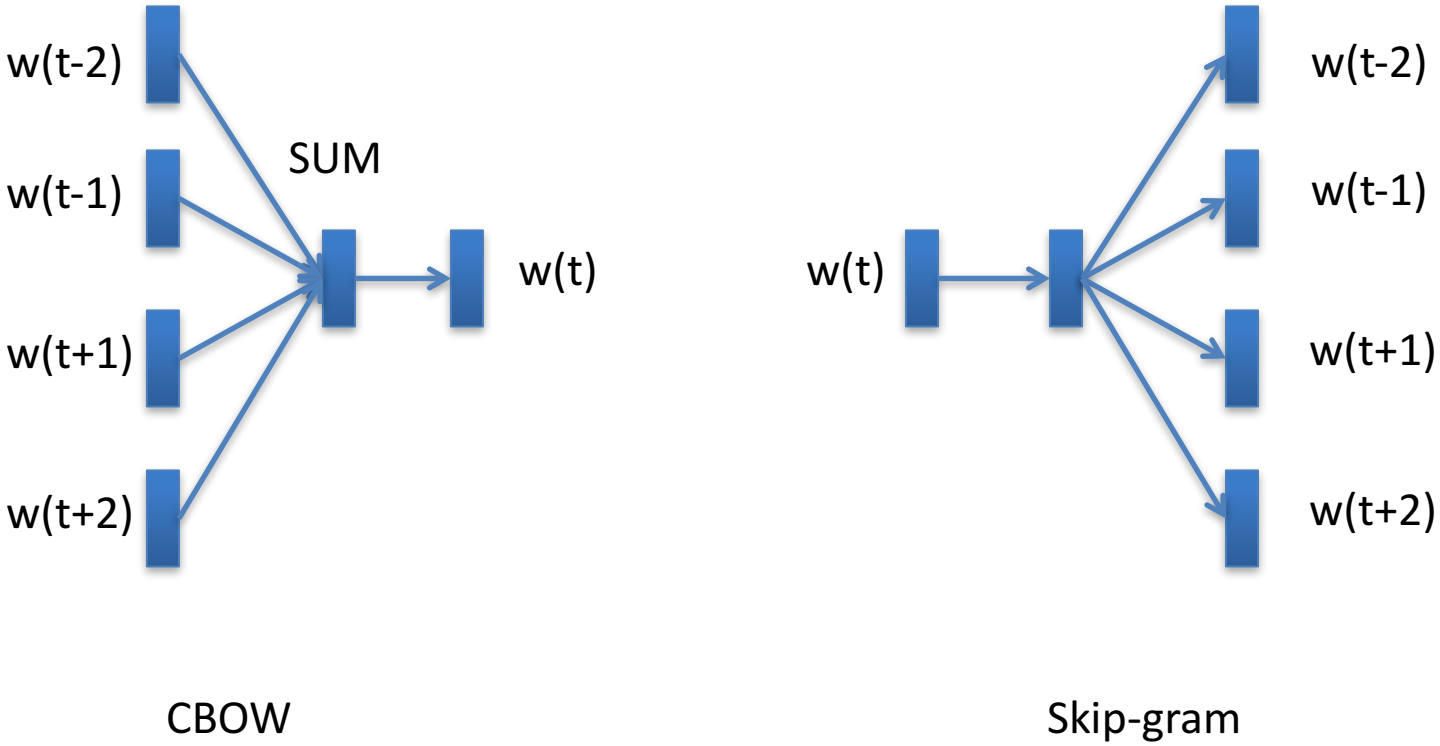
Model	Full	Clean	Noise Type		
			Speech	Music	Other
prod	6.1	5.1	8.5	6.2	6.0
home	5.1	4.9	6.3	5.1	5.0
home(adapt)	4.9	4.7	6.1	4.9	4.8

Table 4: WER on Google Home test set.

Recap: Word Embeddings

Word Embeddings

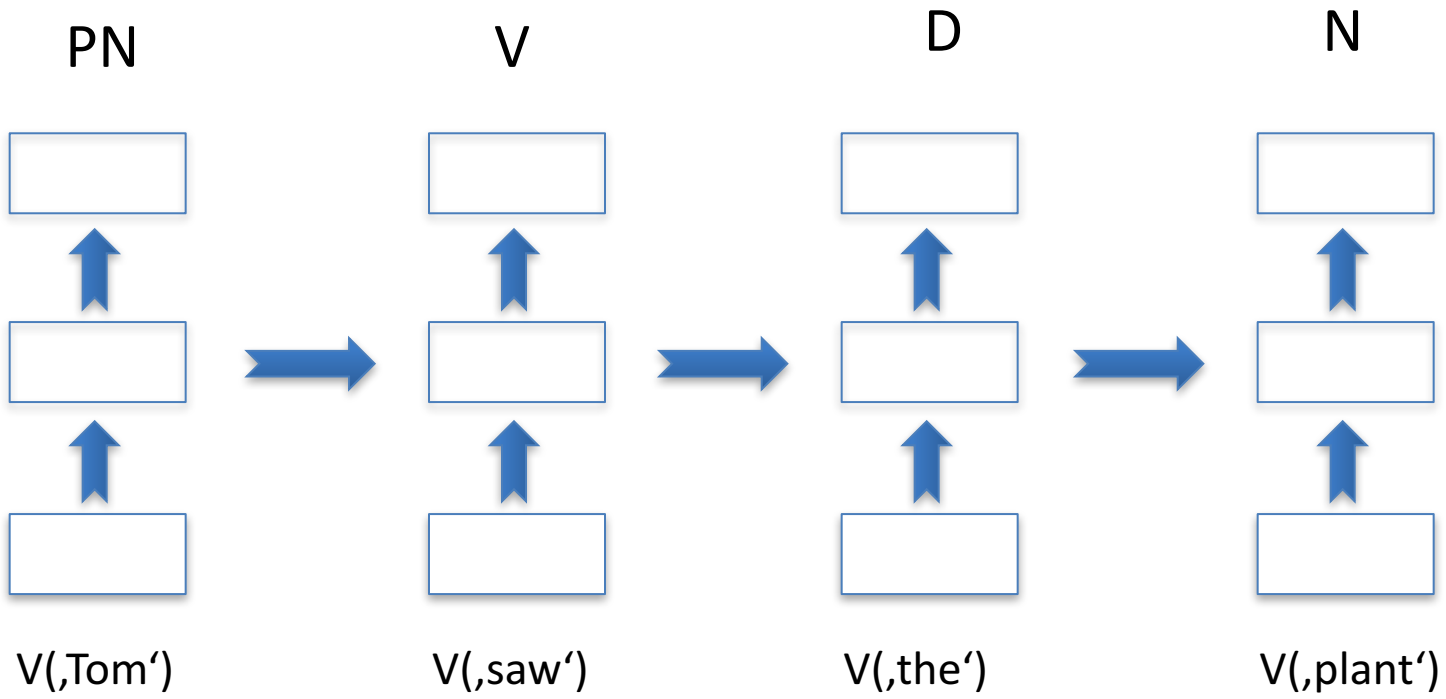
- Another way was proposed by Mikolov et al, (2013)
- https://code.google.com/p/word2vec/



Word Embeddings

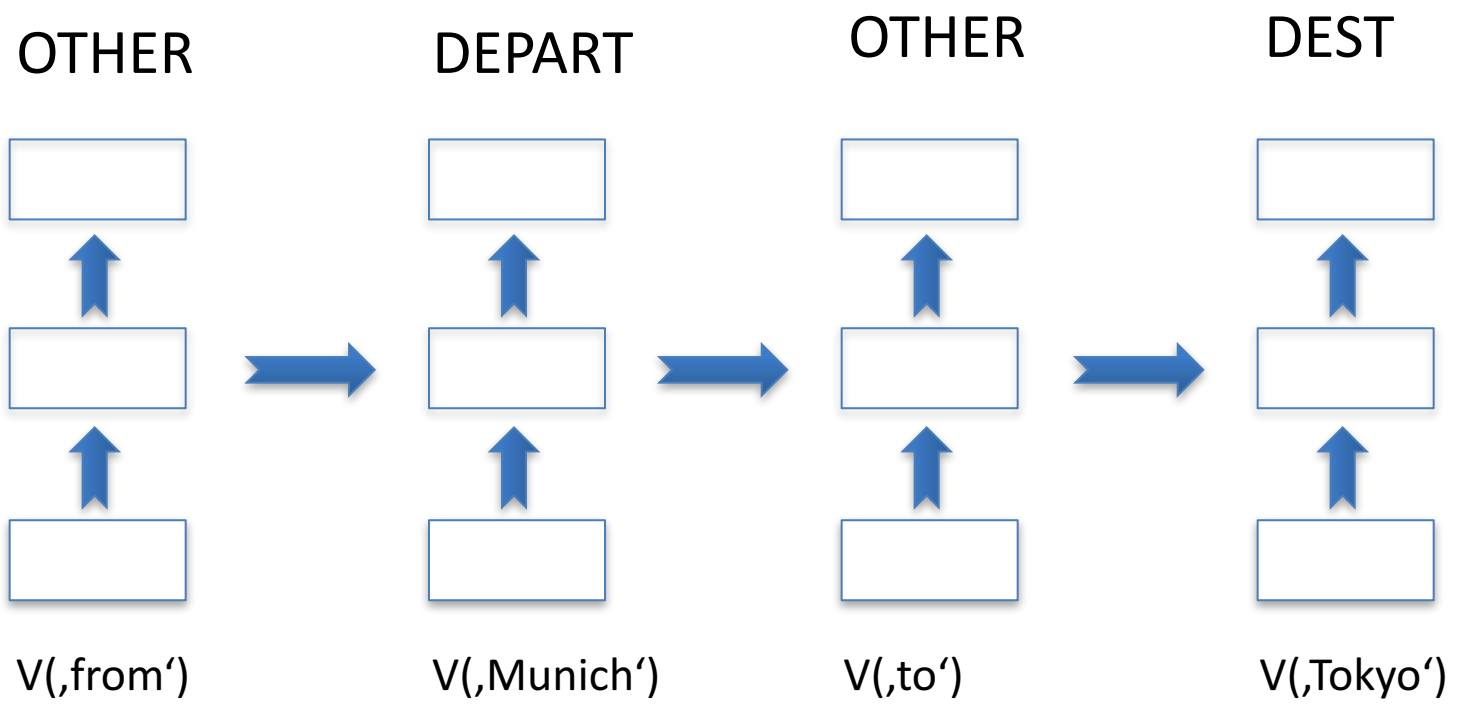
- Linear relations between words:
  - v(king) – v(queen) ≈ v(man) – v(woman)
  - v(japan) – v(tokio) ≈ v(germany) – v(berlin)
  - v(italy) – v(wine) ≈ v(germany) – v(?)
  - v(japan) – v(sushi) ≈ v(germany) – v(?)
- It can be used to solve analogy tasks
  - Rome : Italy ≈ Berlin : ?
  - Compute v(Berlin) – v(Rome) + v(Italy)
  - Find the closest vector

RNN for POS Tagging



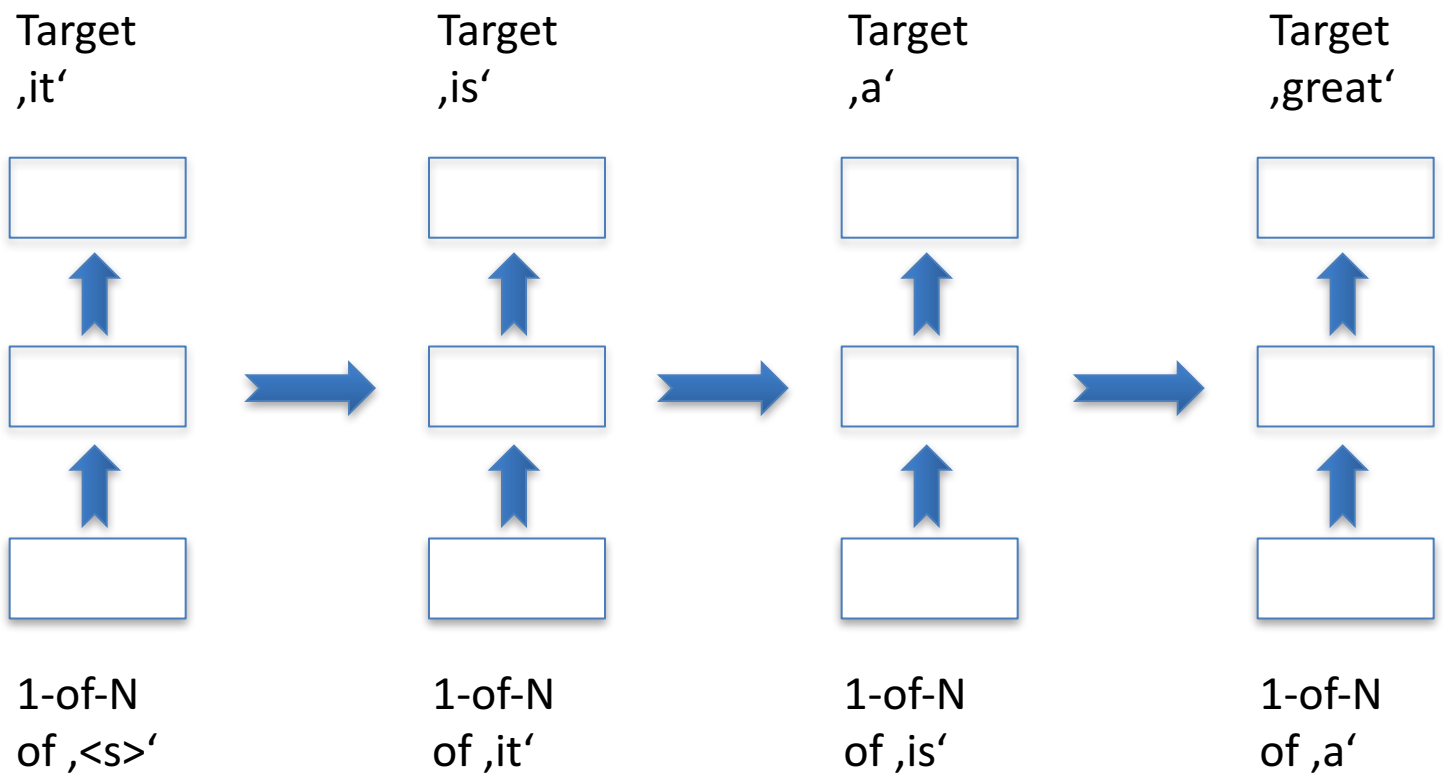
RNN for Slot filling

- e.g. A ticket from Munich to Tokyo please



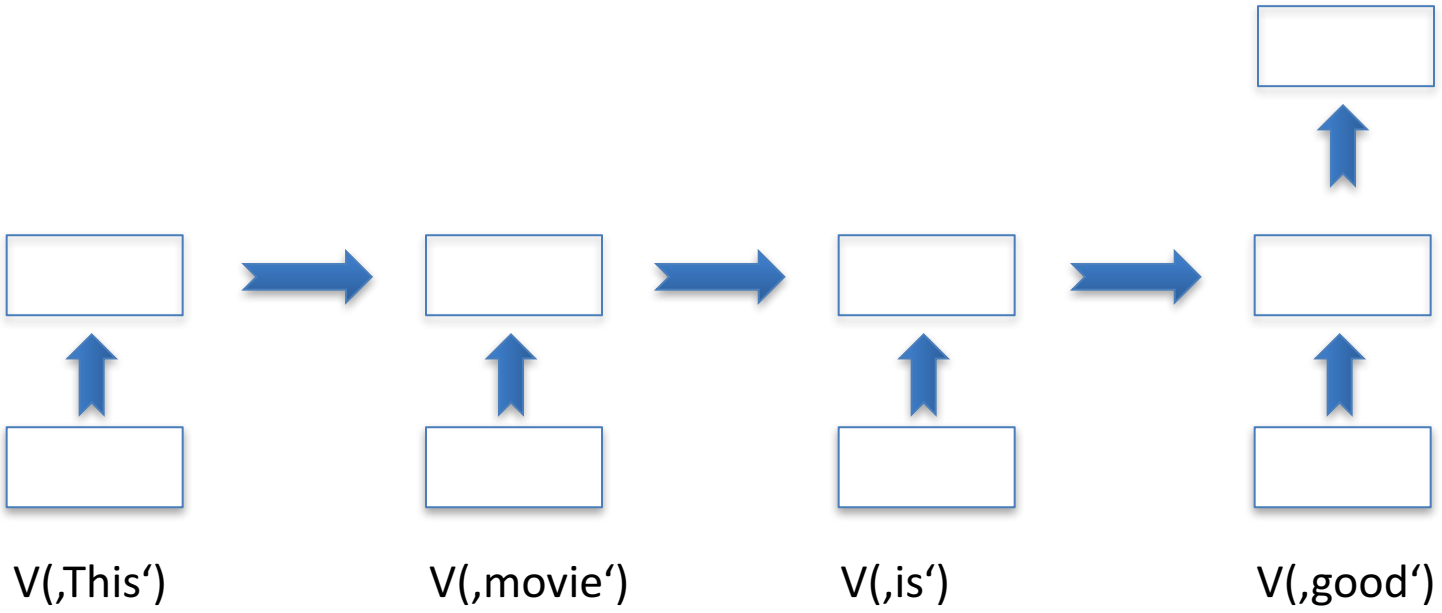
RNN for Language Modelling

- Training data: ‚It is a great day‘



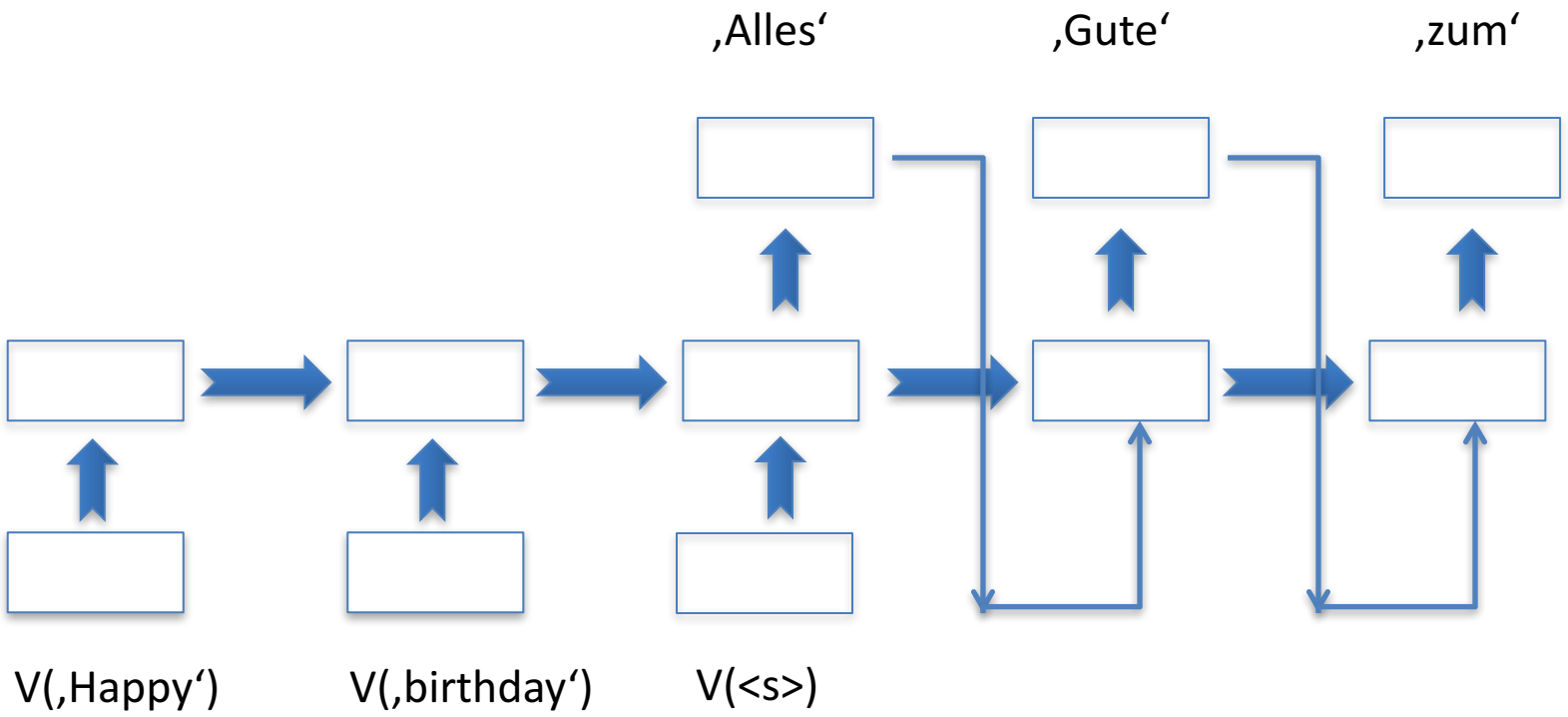
RNN for Sentiment Analysis

- Input is a sentence
- Output is positive, negative or neutral



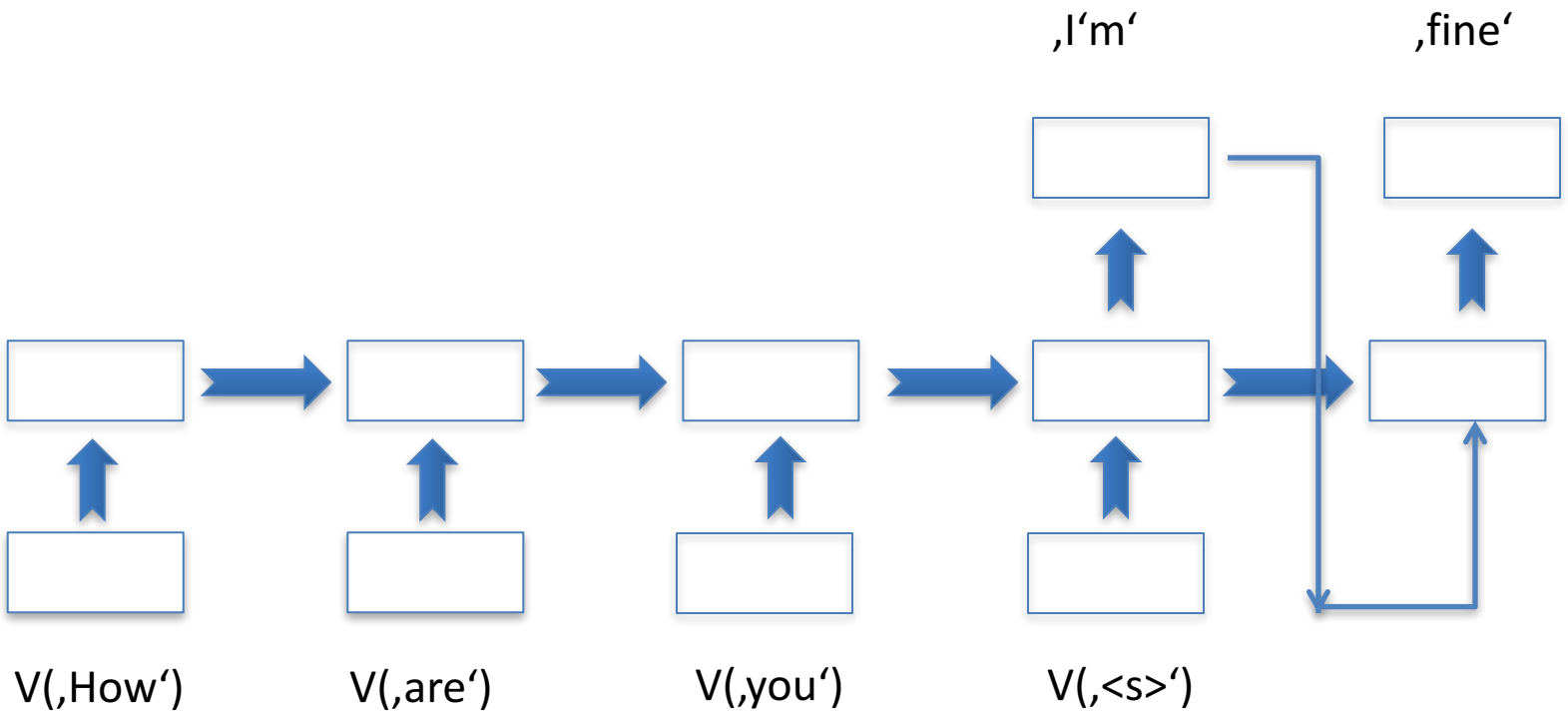
RNN for Machine Translation

- Input is a sentence in the source language
- Output is also a sentence but in the target language



RNN for Chit-chat Dialog Modelling

- Input is fed to the source side
- Output of the system in the target side



Chatbot-style Systems

- Examples:

Machine: *hi*
Human: *hello*
Machine: *could you please let me know what are the operating systems you are using ?*
Human: *i am using linux*
Machine: *ok it was in the ssl vpn on which you stated it was broken*
Machine: *please check if you are logged in to corp network*
Machine: *what is the error that you are running please*
Human: *i am seeing an error related to vpn*
Machine: *what is the error message that you are getting when connecting to vpn using network connect ?*

Machine: *is it a windows machine ?*
Machine: *we have linux on your machine .*
Machine: *is it windows xp or windows 7 ?*
Human: *i am using linux*
Machine: *let me take a remote of your machine and check if issue has been pushed properly .*
Human: *ok*
Machine: *please open terminal and run the below commands in one computer :*
Machine: *sudo apt - get update*
Machine: *sudo apt - get upgrade*
Machine: *sudo <COMMAND> - updater*
Human: *done*
Machine: *let me try connecting now*

Chatbot-style Systems

- Examples:

Human: *hello !*
Machine: *hello !*
Human: *how are you ?*
Machine: *i 'm good .*
Human: *what 's your name ?*
Machine: *i 'm julia .*
Human: *when were you born ?*
Machine: *july 20th .*
Human: *what year were you born ?*
Machine: *1977 .*
Human: *where are you from ?*

Human: *what is the purpose of life ?*
Machine: *to serve the greater good .*
Human: *what is the purpose of living ?*
Machine: *to live forever .*
Human: *what is the purpose of existence ?*
Machine: *to find out what happens when we get to the planet earth .*

Thanks for listening!