Fpt Software | CAMPUS

NATURAL LANGUAGE PROCESSING

May 21,
2019

Agenda

1. Technology Landscape
2. Overview
3. Customer References
4. Key Service Offerings
5. Human Resources
6. Case Studies

NLP - INTRODUCTION

What is NLP?

Natural Languages

- English, Mandarin, French, Swahili, Arabic, Nahuatl, etc.
- NOT Java, C++, Perl, Python, etc.

Ultimate Goal

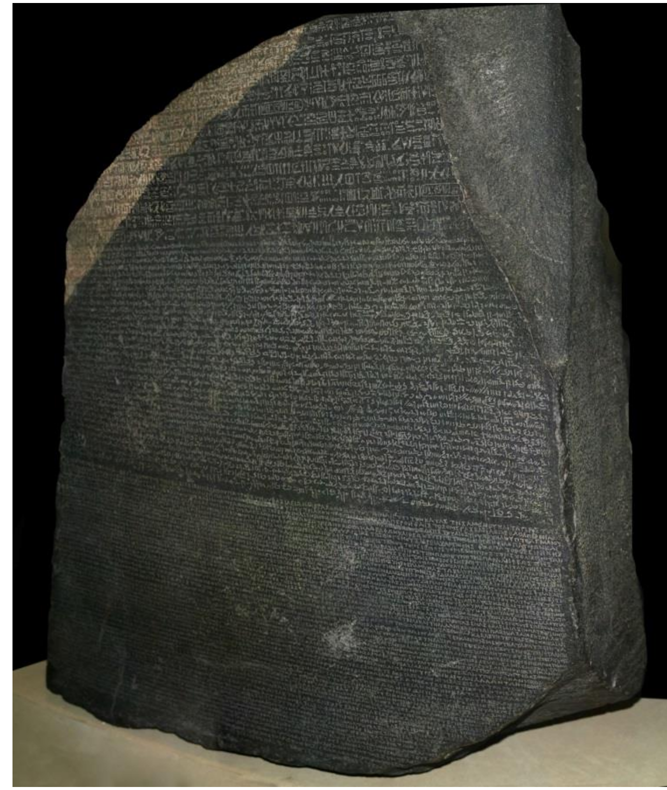
- Natural human-to-computer communication

Sub-field of A.I but very interdisciplinary

- Computer science, human-computer interaction (HCI), linguistics, cognitive psychology, speech signal processing (EE), etc.

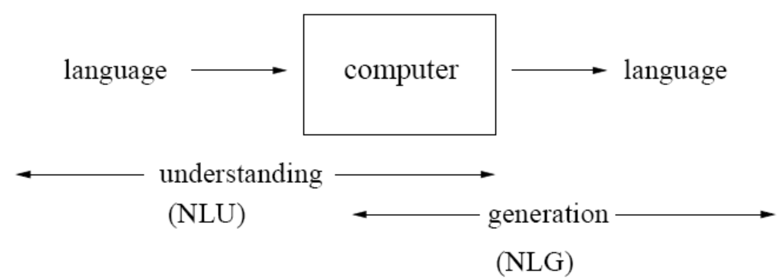
Applications

- Speech recognition and synthesis
- Machine translation
- Document processing
  - Information extraction
  - Summarization
- Text generation
- Dialog systems (typed and spoken)



Rosetta Stone

computers using natural language as input and/or output



How does NLP work…

Morphology: What is a word?

- **奧林匹克運動會**（希臘語：**Ολυμπιακοί Αγώνες**，簡稱**奧運會**或**奧運**）是國際奧林匹克委員會主辦的包含多種體育運動項目的國際性運動會，每四年舉行一次。
- كبيوتها = “to her houses”

Lexicography: What does each word mean?

- He plays bass guitar.
- That bass was delicious!

Syntax: How do the words relate to each other?

- The dog bit the man. ≠ The man bit the dog.
- But in Russian: человек собаку съел = человек съел собаку

How does NLP work…

Semantics: How can we infer meaning from sentences?

- I saw the man on the hill with the telescope.

- The ipod is so small! ☺
- The monitor is so small! ☹

Discourse: How about across many sentences?

- President Bush met with President-Elect Obama today at the White House. He welcomed him, and showed him around.

- Who is “he”? Who is “him”? How would a computer figure that out?

How does NLP work…

- **Phonology**:
  - What words (or sub words) are we dealing with?
  - sounds -> words
    - /b/ + /o/ + /t/ = boat
- **Morphology**: How words are constructed from more basic meaning units?
  - morphemes -> words
    - friend + ly = friendly
- **Syntax**: What phrases are we dealing with?
  - word sequence -> sentence structure
- **Semantics**: What's the context-free meaning?
  - sentence structure + word meaning -> sentence meaning
- **Pragmatics**: What is the more exact (context-dependent) meaning?
  - sentence meaning + context -> more precise meaning
- **Discourse Knowledge**: How the immediately preceding sentences affect the interpretation of the next sentence?
  - Discourse and world knowledge
- **World knowledge**: Using general knowledge about the world

Spoken Language Processing

Related things

- Making computer voices sound more human
- Making computer speech acts more human-like

Why is NLP hard?

- Requires both understanding the “from” language and generating the “to” language.
- How can we teach a computer a “second language” when it doesn’t even really have a first language?
- Can we do machine translation without solving natural language understanding and natural language generation first?

	What hunger have I
Que hambre tengo yo	*I've got that hunger*
	I am so hungry
She let the cat out of the bag.	*Ella deja que el gato fuera de la bolsa*

NLP – Speech to Text

What is Automatic Speech Recognition?

Automatic speech recognition (ASR) is an important technology to enable and improve the human-human and human-computer interactions.

Goal is to generate transcription from audio signal with best accuracy possible

Speech Recognition Problem

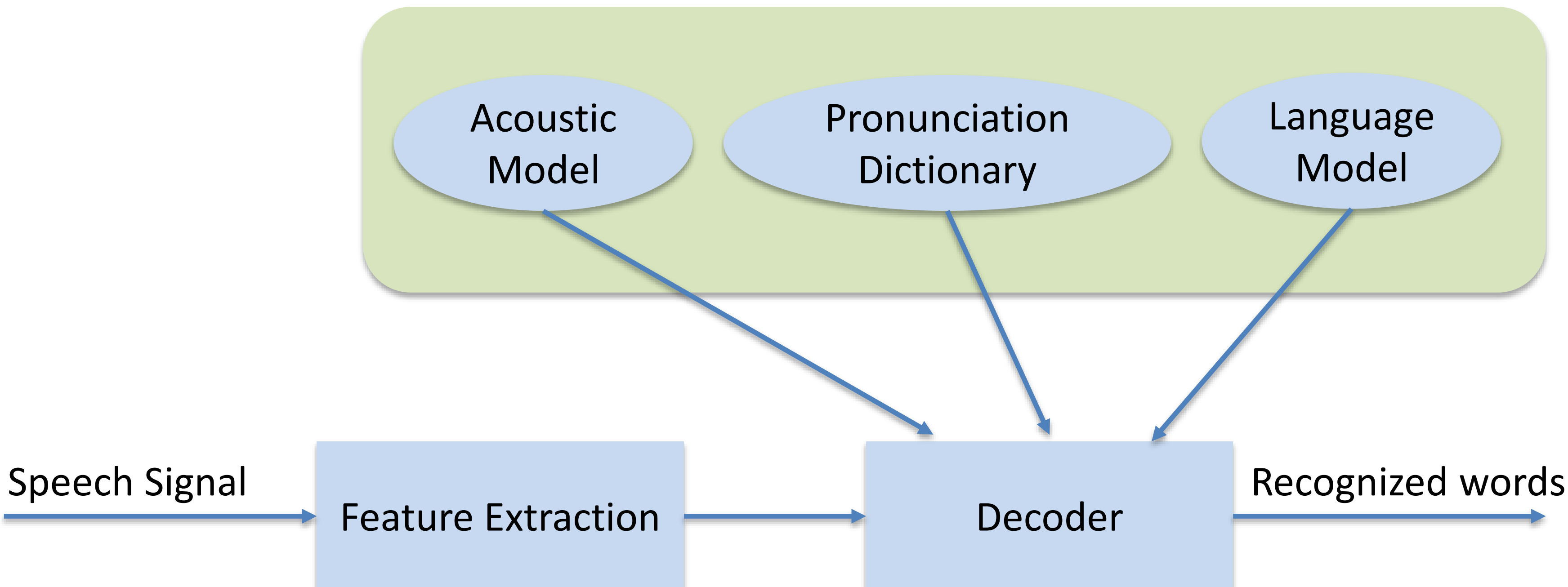
Audio Waveform

$$\underset{W}{\arg\max}\ p(W|O) = \underset{W}{\arg\max}\ \underbrace{p(O|W)}_{\text{Acoustic Model}}\ \underbrace{p(W)}_{\text{Language Model}}$$

Word Sequence

* Slide from V. Vanhoucke, ICML 2013 Keynote

Conventional Automatic Speech Recognition Model



A typical system architecture for ASR

Four Generations of ASR

- **1st generation (1950s - 1960s)**
  Work based on acoustic-phonetics
- **2nd generation (1960s - 1970s)**
  ASR based on template-matching
- **3rd generation (1970s - 2000s)**
  ASR based on statistical model (HMM, GMM)
- **4th generation (late 2000 - present)**
  ASR model based on deep learning

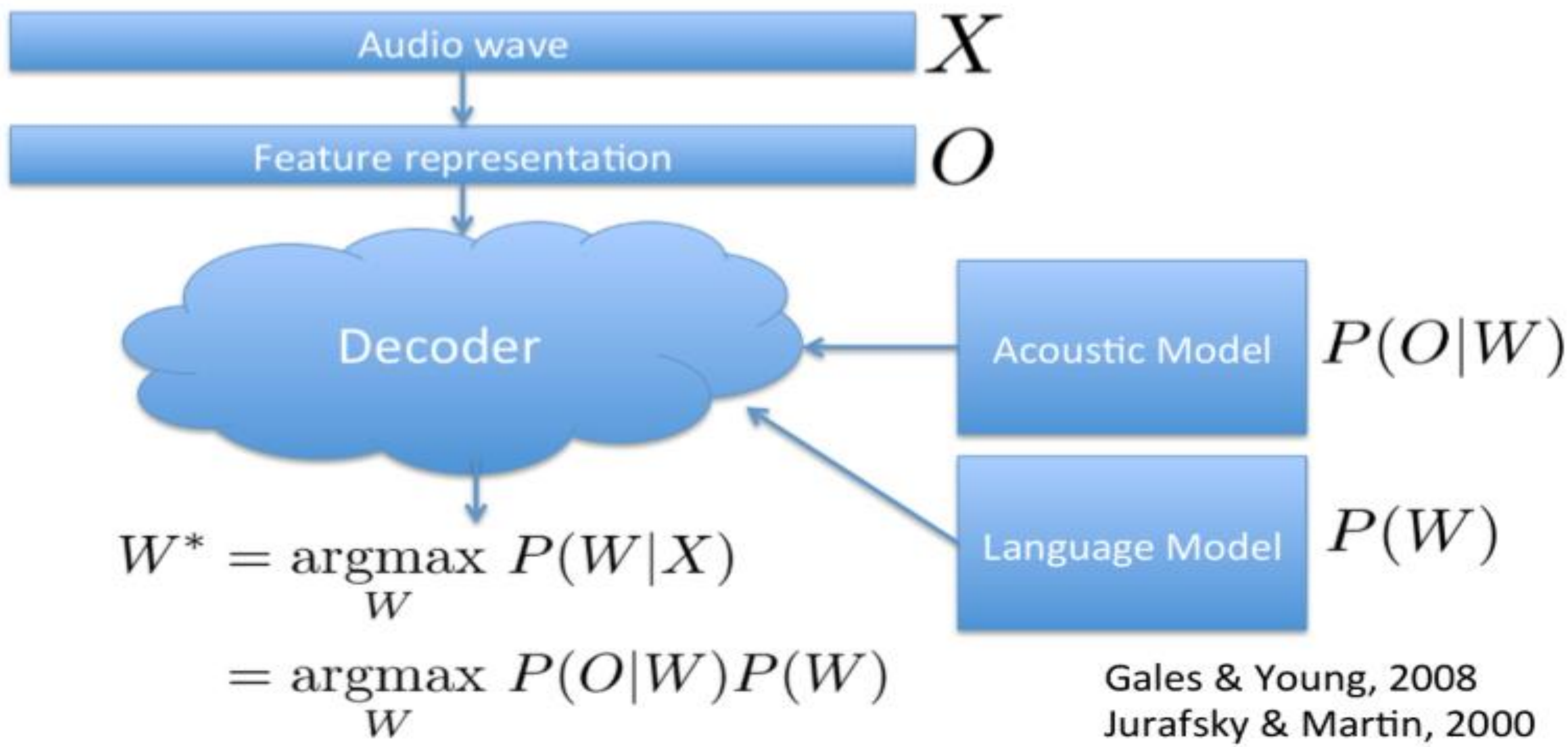
Traditional ASR pipeline

- Usually represent words as sequence of "phonemes"

  For example, w1 = "hello" = [hh eh l ow] = [q1q2q3q4]
- Phonemes are perceptually distinct units of sounds that distinguish words. For i.e., refer to the following table, TIMIT original phone set.

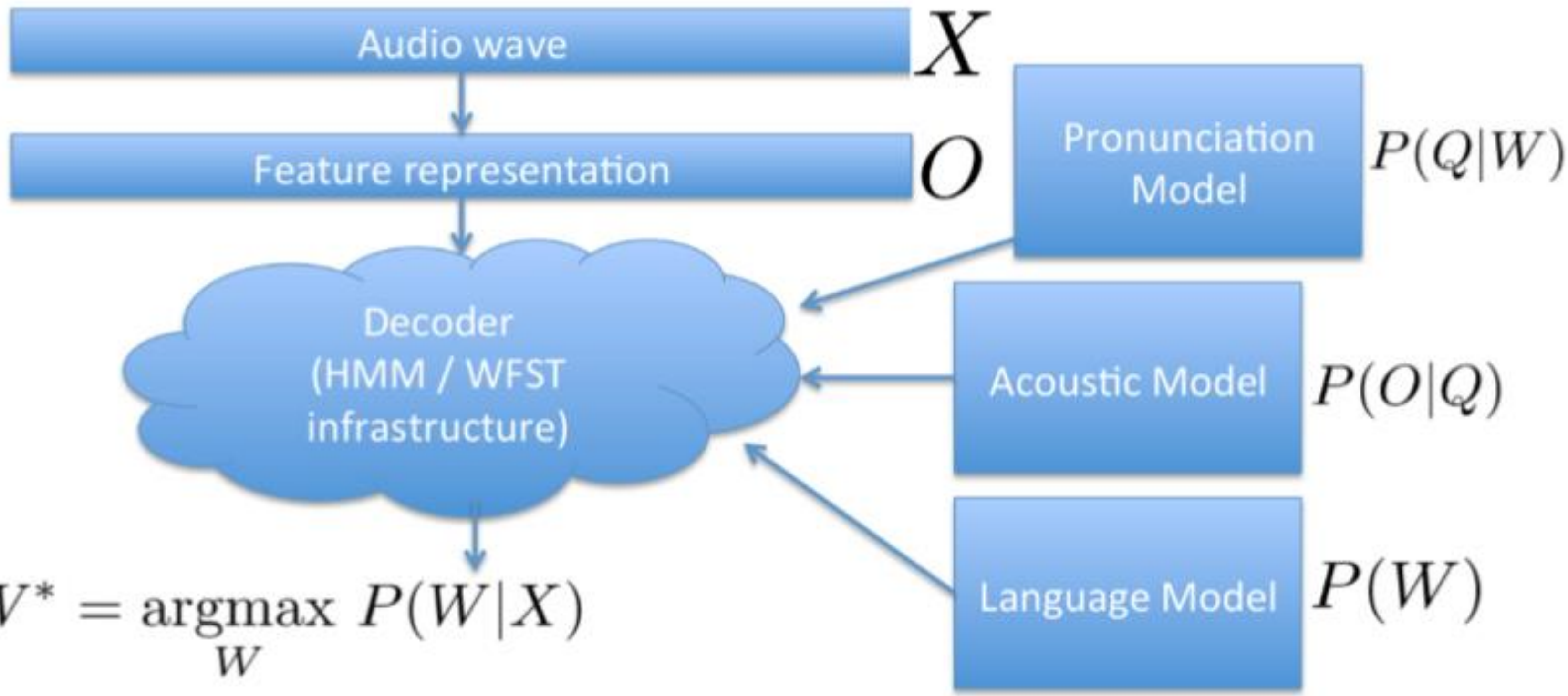
	Phone Label	Example		Phone Label	Example		Phone Label	Example
1	iy	beet	22	ch	choke	43	en	button
2	ih	bit	23	b	bee	44	eng	Washington
3	eh	bet	24	d	day	45	l	lay
4	ey	bait	25	g	gay	46	r	ray
5	ae	bat	26	p	pea	47	w	way
6	aa	bob	27	t	tea	48	y	yacht
7	aw	bout	28	k	key	49	hh	hay
8	ay	bite	29	dx	muddy	50	hv	ahead
9	ah	but	30	s	sea	51	el	bottle
10	ao	bought	31	sh	she	52	bcl	b closure

Traditional ASR pipeline



Gales & Young, 2008
Jurafsky & Martin, 2000

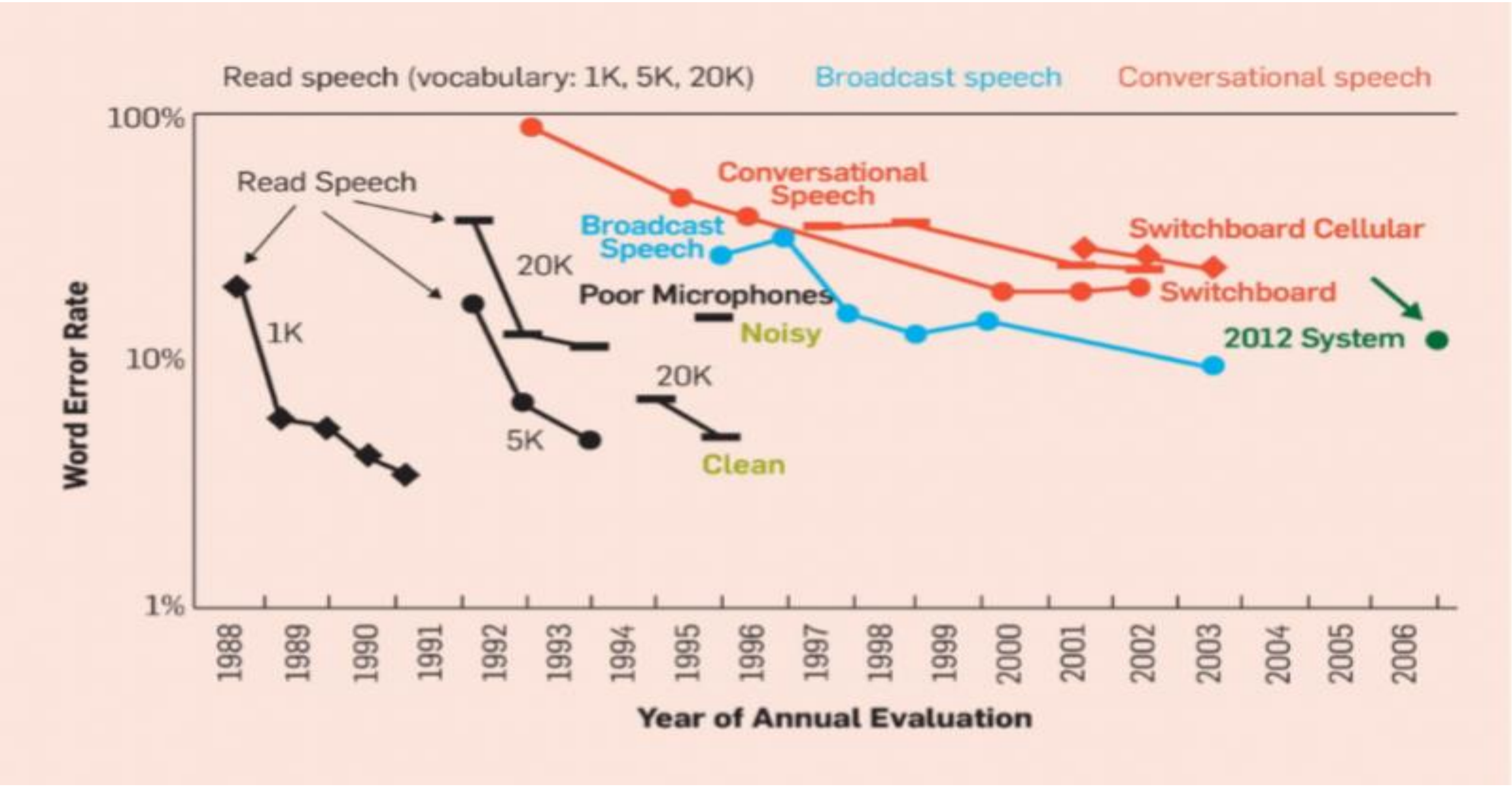
Traditional ASR pipeline



$$W^* = \operatorname*{argmax}_W P(W|X)$$

$$= \operatorname*{argmax}_W \sum_Q P(O|Q)P(Q|W)P(W)$$

Progresses Made Before Deep Learning



*X. Huang, J Baker, Reddy, “A historical perspective of speech recognition”

We are now in the 4th generation of ASR empowered by deep learning technologies..

- **Deep Acoustic Modeling**
  - Deep feedforward neural networks (DNNs, CNNs…)
  - Deep recurrent neural networks (LTSM, GRU…)
  - End-to-end neural networks

- **Deep Language Modeling**
  - Deep feedforward neural networks (DNNs, CNNs…)
  - Deep recurrent neural networks (RNNs, LTSM…)
  - Word embedding

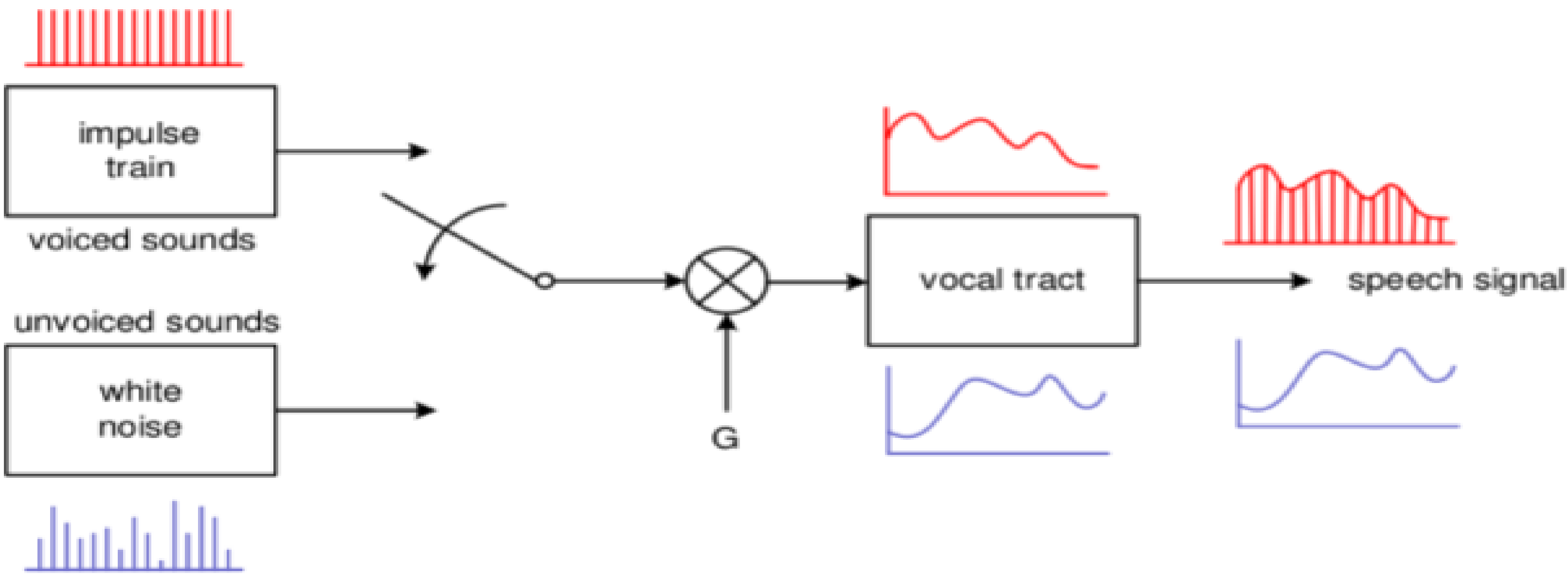
End-to-end Speech Recognition System

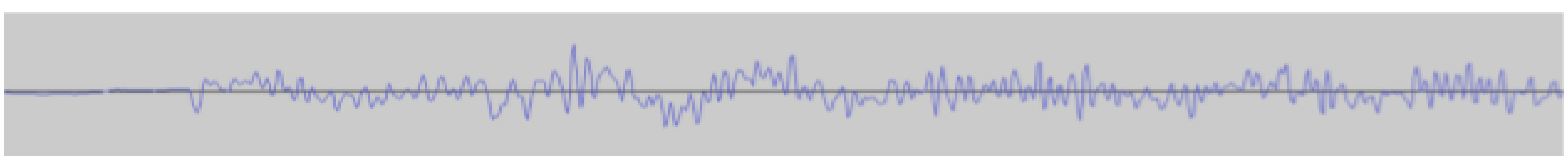
End-to-end Speech Recognition

- Preprocessing: feature extraction from time to frequency domain
- Network structure: RNN or BRNN
- LTSM cells in the network
- CTC for decoding the transcript from temporal RNN output
- Training
- Language models

Preprocessing

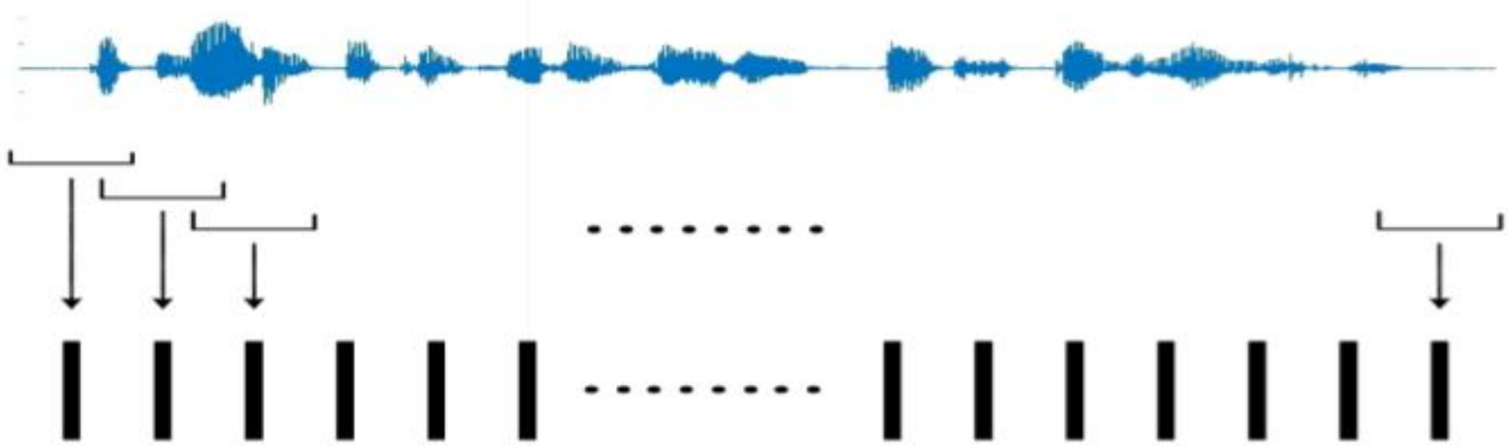
Feature extraction from time to frequency domain





Preprocessing

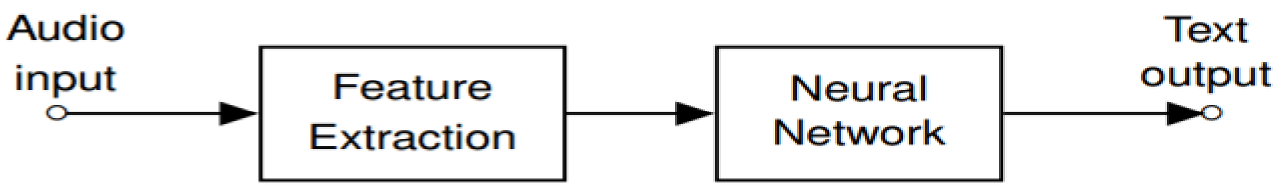
Feature extraction from time to frequency domain

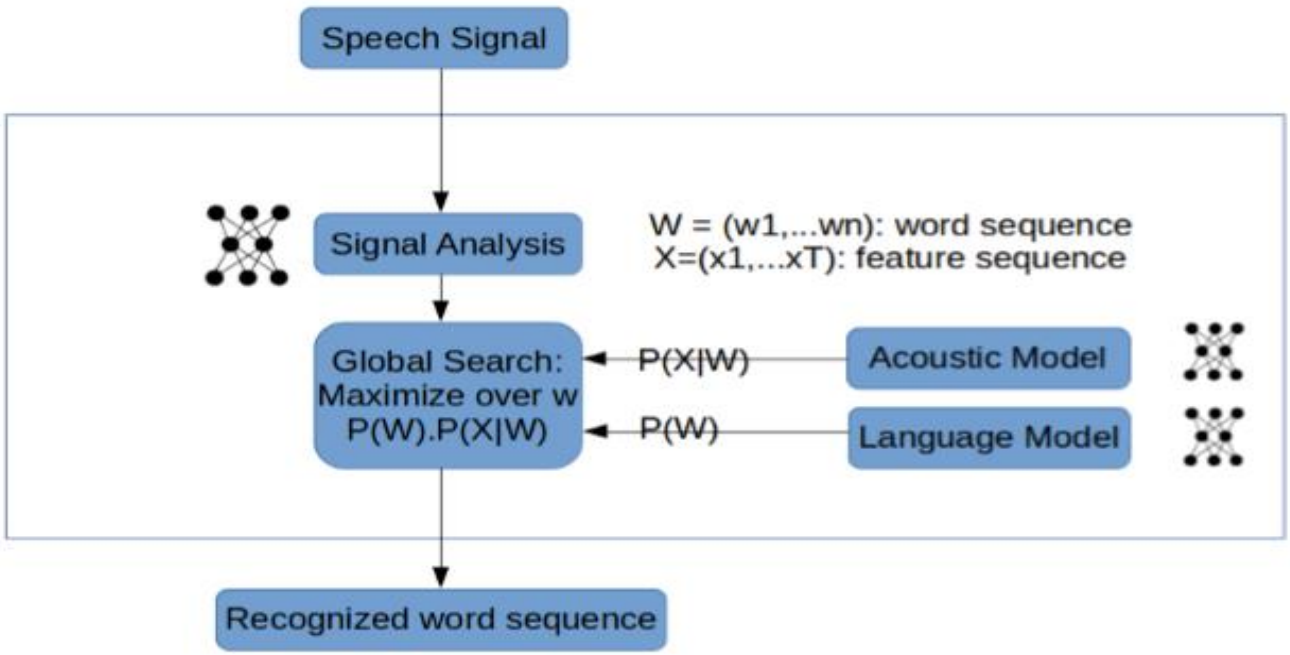


- Frame window length 10ms – 20ms: take a small window of waveform
- Compute FFT and take magnitude (i.e., power)
- Describe frequency content in local window
- Commonly used hand-crafted features:
  - Linear Predictive Cepstral Coefficients (LPCCs)
  - Mel-frequency Cepstral Coefficients (MFCCs)
  - Perceptual Linear Predictive (PLP) analysis
  - Mel-frequency Filter bank

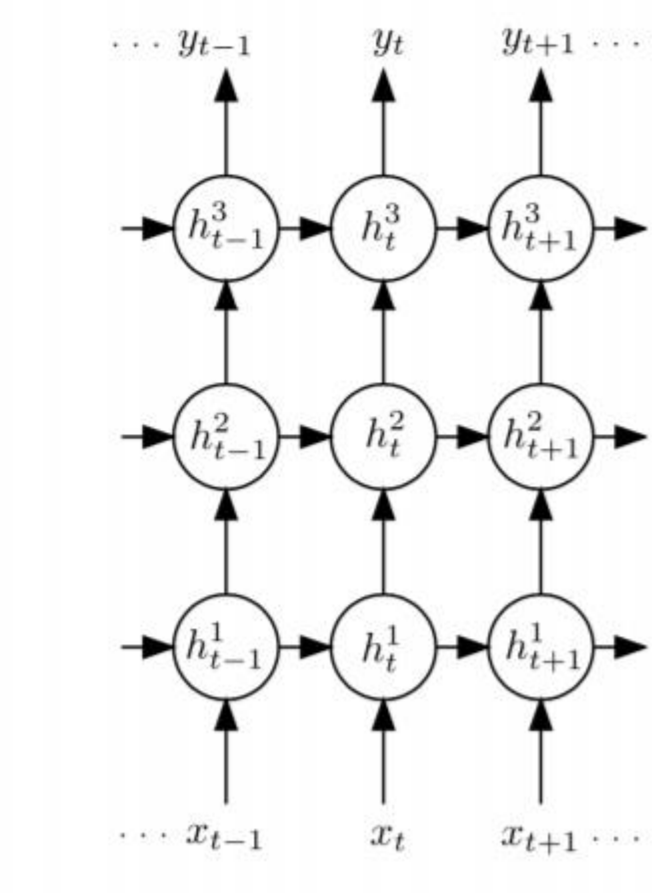
End-to-end Speech Recognition System

Block diagram of a generic end-to-end speech recognizer





Recurrent Neural Network - RNN



End-to-end Speech Recognition System

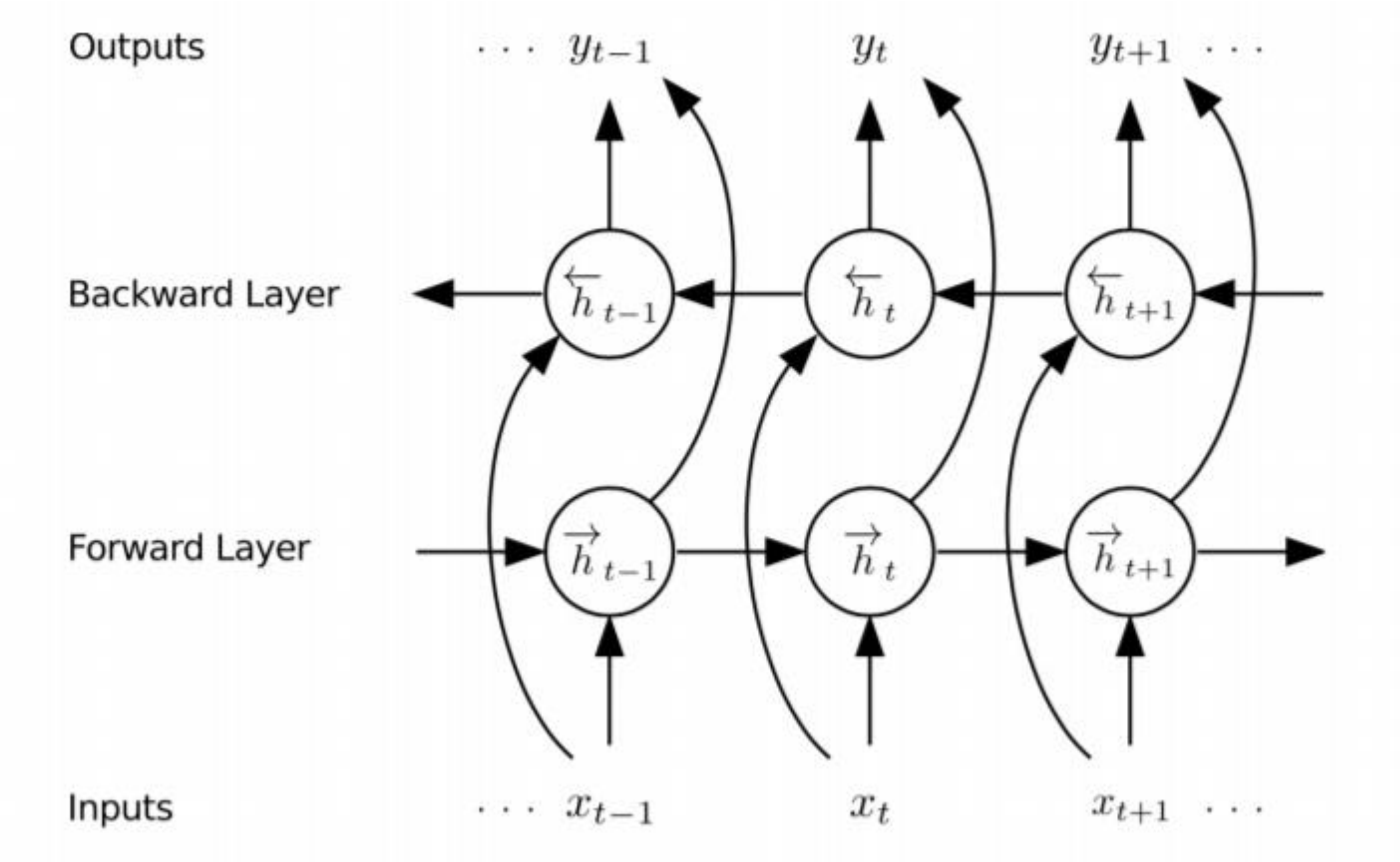
Recurrent Neural Network - RNN

- Output of hidden layer and output layer:

$$h = f(W_{ih}x_t + W_{hh}h_{t-1} + b_h)$$
$$y = W_{ho}h_t + b_o$$

- Fully connected hidden layers
- Connection from previous outputs of the hidden layer to the input of the layer

Bidirectional RNN - BRNN

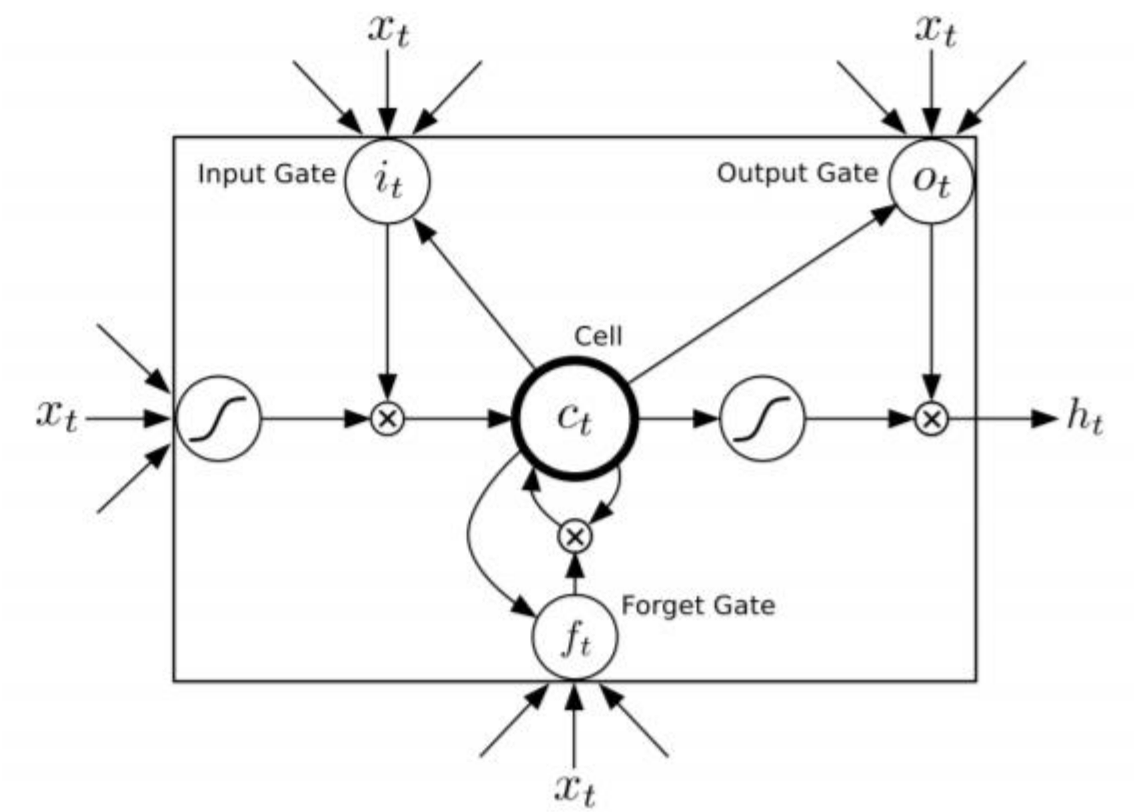


End-to-end Speech Recognition System

Bidirectional RNN - BRNN

- Relational to past and future
  - Speech is recognized usually one utterance at once
  - This allows to use also future relations
- The outputs are calculated similarly to RNN
  - Both directions separately
  - Combined in the output layer
- Can be though as a two parallel networks where each network has inverse time relations compared to other.

Long Short-term Memory - LTSM



LTSM Cell

LTSM Cell

- Cell enables longer relations in time for RNN in comparison to Perceptrons
- Forget gate to remove all the futile memory
- Equations for input, forget, and output gates cells

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

CTC

- RNN generate temporal results -> CTC to get transcripts from temporal labels
- Trained with CTC to get distributions of the possible alignments
- The decoding of CTC is done with forward-backward algorithm by summing all probabilities of all possible alignments.

Word Error Rate

- Error measure used widely in ASR research
- Take account deletions, insertions and substitutions
- Example:
  - It’ e news, false and fake
  - It’s all tall(I) fake shoes(S), false (D) fakes (S)
- WER = $\frac{N_d+N_i+N_s}{N_{all}}$

Results From Papers

Table 1. **Wall Street Journal Results.** All scores are word error rate/character error rate (where known) on the evaluation set. 'LM' is the Language model used for decoding. '14 Hr' and '81 Hr' refer to the amount of data used for training.

SYSTEM	LM	14 HR	81 HR
RNN-CTC	NONE	74.2/30.9	30.1/9.2
RNN-CTC	DICTIONARY	69.2/30.0	24.0/8.0
RNN-CTC	MONOGRAM	25.8	15.8
RNN-CTC	BIGRAM	15.5	10.4
RNN-CTC	TRIGRAM	13.5	8.7
RNN-WER	NONE	74.5/31.3	27.3/8.4
RNN-WER	DICTIONARY	69.7/31.0	21.9/7.3
RNN-WER	MONOGRAM	26.0	15.2
RNN-WER	BIGRAM	15.3	9.8
RNN-WER	TRIGRAM	13.5	8.2
BASELINE	NONE	—	—
BASELINE	DICTIONARY	56.1	51.1
BASELINE	MONOGRAM	23.4	19.9
BASELINE	BIGRAM	11.6	9.4
BASELINE	TRIGRAM	9.4	7.8
COMBINATION	TRIGRAM	—	6.7

Language Models

- Standard approach: n-gram model
  - Simple n-gram models are common
  - Depends on specific domain -> This is a disadvantage point of using n-gram

- Neural network language model

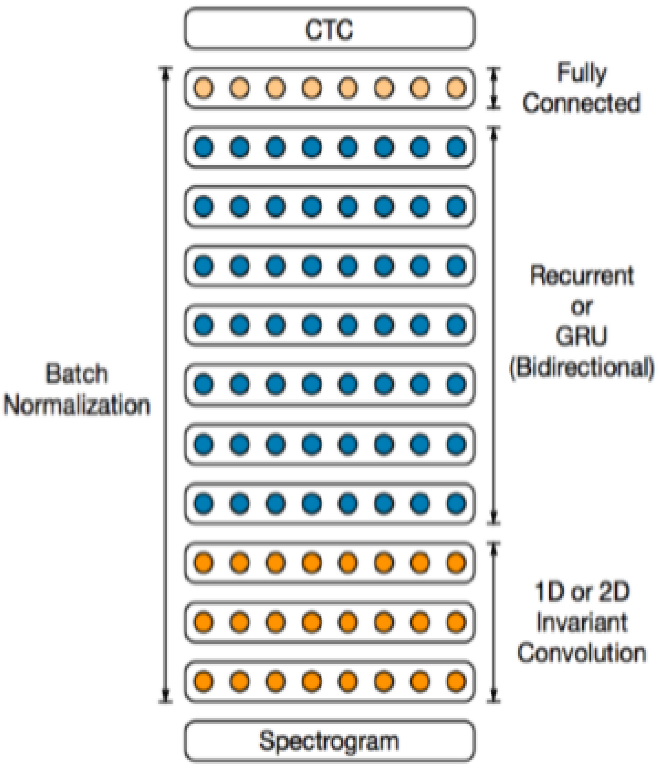
- Combine N-gram and deep neural networks.

Lexicon and Decoder

- Lexicon
  - Related to linguistics
- Decoder
  - Use Viterbi algorithm
  - Viterbi is a dynamic programming algorithm for finding the most likely sequence of hidden states.
  - More on Viterbi: http://salve.twiki.di.uniroma1.it/pub/NLP/WebHome/Tutorial Viterbi.pdf

Baidu's Deep Speech 2

- ➢ The architecture
- RNN to predict graphemes (26 characters, space, blank)
  - ➢ Spectrograms as input
  - ➢ 1 layer of convolutional filters
  - ➢ 3 layers of Gated Recurrent Units (1000 neural per layer)
  - ➢ 1 full-connected layer to predict c
  - ➢ Batch normalization
- CTC loss function (Warp-CTC)
- Trained with SGD + Nesterov momentum



Training data for ASR

- LibriSpeech
  - http://www.openslr.org/12
  - Read speech from public domain audiobooks

- Wall Street Journal
  - htpps://catalog.ldc.upenn.edu/ldc93s6a
  - Reading WSJ articles

Frameworks to build ASR. system

- Open Source
  - CMU Sphinx: Traditional ASR model, no neural network
  - Kaldi ASR: support DNNs, RNN, LSTM-DNNs
  - TensorFlow: support DNNs, RNN, LSTM-DNNs
  - Wit.ai
  - Neon by Nervana Intel

- Cloud service
  - Google Cloud Speech API
  - IBM Watson Speech API
  - Microsoft Bing API
  - Amazon Lex

Exercise

- What are the benefits and disadvantages of end-to-end speech recognition compared to conventional speech recognition?

NLP – Text to Speech

Text to Speech - Introduction

Definition

- A Text-To-Speech (TTS) System is a computer-based system that automatically converts text into artificial human speech.
- TTS synthesizers do not playback recorded speech; rather, they generate sentences using plain text as input.

TTS synthesizer vs Voice Response System

- It is necessary to distinguish between TTS synthesizer from Voice Response Systems. Voice Response Systems simply concatenate words and segments of sentences and are applicable only in situations where limited vocabulary is requires, and pronunciation restrictions exist.

Speech synthesizer being considered in this context actually excapsulate models of the human vocal tract to produce a synthetic human voice output corresponding to input text.

- Since it is impracticable to store pre-recorded audio clips of all words of a language, automatic “Pronunciation” of words and sentences is necessary in TTS systems.

The Text-To-Speech (TTS) Problem

- TTS system processes are significantly different from live human speech production (and language analysis). Live human speech production depends of complex fluid mechanics dependent on changes in lung pressure and vocal tract constrictions. Designing systems to mimic those human constructs would result in avoidable complexity

- Process of converting written text into spee

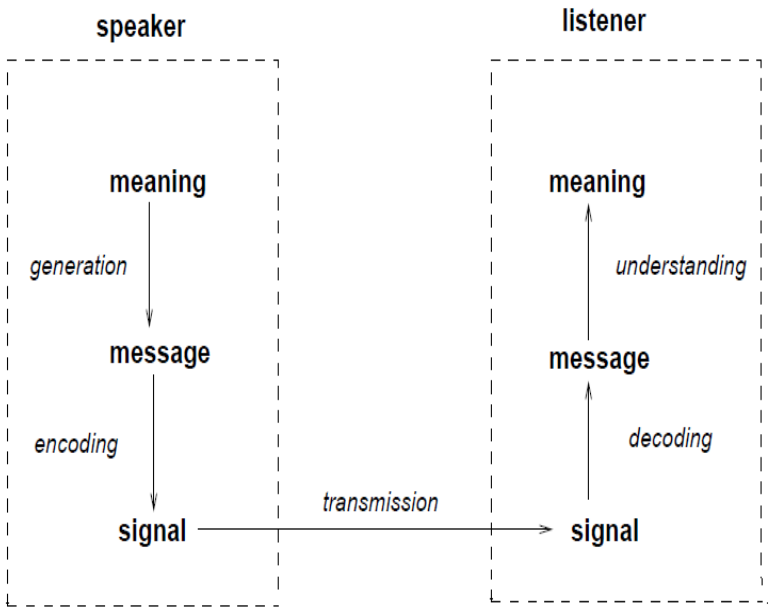
- Dealing with Speech:

Speech encode → speech synthesis

Speech decode → Speech recognition

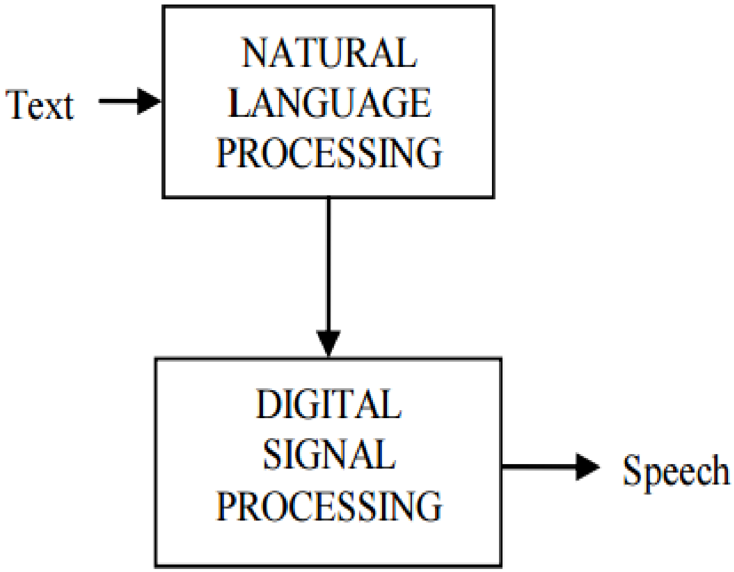
- Dealing with Writing:

Text decode → Text analysis



The Text-To-Speech (TTS) Problem

In general terms, a TTS synthesizer comprises of two parts:

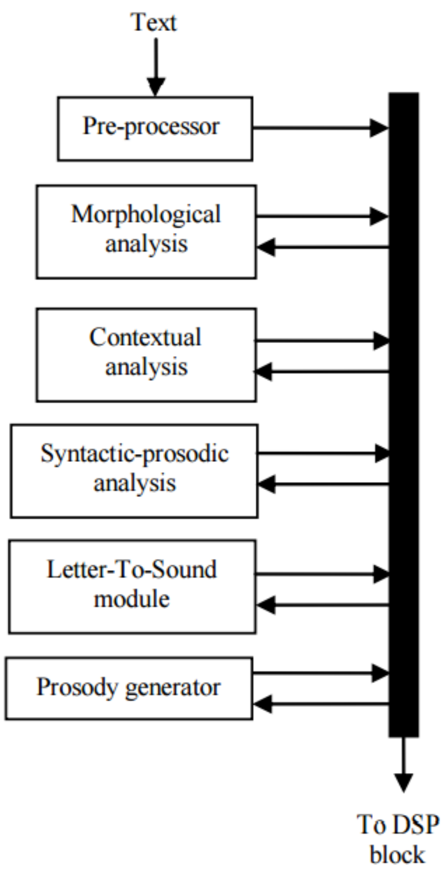


- Natural Language Processing (NLP)

- Digital Signal Processing (DSP)

The Nature Language Processing (NLP)

The NLP module comprises of text analyter, phonetization and prosody generation module.



Text analyzer: Comprises of four basic parts: pre-processing block, morphological analysis block, contextual analysis block and syntactic-prosodic parser.

- Pre-processing block converts abbreviations, numbers and acronyms into full text when necessary.

- Morphological analysis block categorizes each word in the sentence being analysis into possible parts of speech.

- Contextual analysis module streamlines the list of possible parts of speech of words in sentences, by considering the parts of speech of neighbouring words.

- Syntactic-prosodic parser locates the text structure.

Phonetization: A Letter-To-Sound (LTS) module is used for phonetic transcription of incoming text.

Prosody generation: Refers to rhythm, stress and intonation of speech.

The Digital Signal Processing (DSP)

The DSP components handles the actual machine “Pronunciation” of words, phrases and sentences, analogous to human speech articulation. This component can be implemented in two ways, namely rule-based synthesis and concatenative synthesis.

Rule-based synthesizers: (usually formant synthesizers)

- generate speech via the dynamic modification of several parameters. Such as: fundamental frequency, voicing and noise levels are modified over time to create an artificial speech waveform.

- Many formant-based speech synthesis systems generate unnatural speech (not sounding human)

Concatenative synthesizers:

- string together pieces of recorded speech extracted from a database of speech samples.

- As a result, concatenative synthesizer generate the most natural-sounding artificial speech.

Solution

WaveNet: is a deep neural network for generating raw audio created by researchers at London-based artificial intelligence firm DeepMind. The technique, outlined in a paper in September 2016, is able to generate realistic-sounding human-like voices by sampling real human speech and directly modelling waveforms.

Deep Voice: a production-quality text-to-speech system constructed entirely from deep neural networks.

- Deep Voice uses Deep Learning for all pieces of the TTS pipeline. Previous TTS systems used Deep Learning for different components of the pipeline but no previous work has gone so far as to replace all major components with Neural Networks.

- It requires very little features engineering and is hence easy to apply to different datasets. Through using Deep Learning, the authors are able to avoid a large amount of feature processing and engineering compared to traditional pipelines.

- It is extremely fast compared to the state of the art and is designed to be used in production systems.

Deep Voice – The Inference Pipeline

The TTS system consists of five major building blocks:

The Text-To-Phoneme Model converts from written text (English characters) to phonemes (encoded using a phonemic alphabet such as ARPABET).

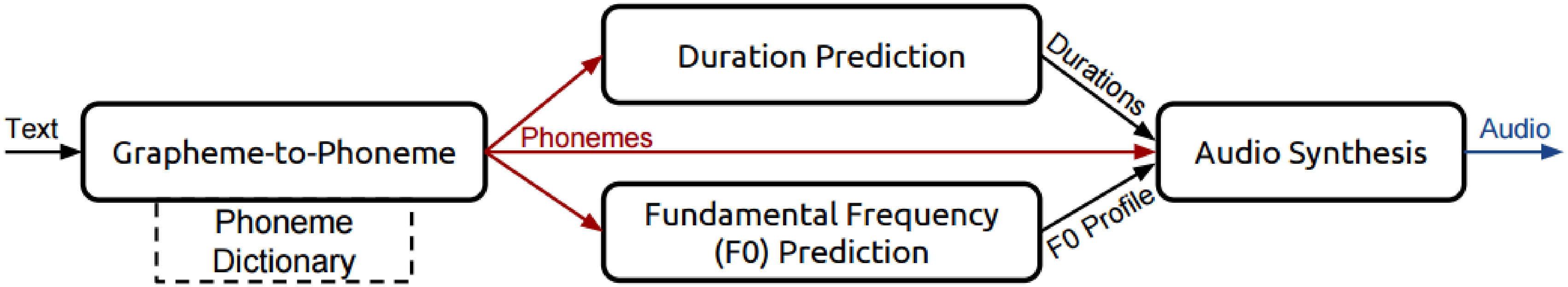
The segmentation model locates phoneme boundaries in the voice dataset. Given an audio file and a phoneme-by-phoneme transcription of the audio, the segmentation model identifies where in the audio each phoneme begins and ends.

The phoneme duration model predicts the temporal duration of every phoneme in a phoneme sequence (an utterance).

The fundamental frequency model predicts whether a phoneme is voiced. If it is, the model predicts the fundamental frequency (F0) throughout the phoneme's duration.

The audio synthesis model combines the outputs of the grapheme-to-phoneme, phoneme duration, and fundamental frequency prediction models and synthesizes audio at a high sampling rate, corresponding to the desired text.

Deep Voice



Through this pipeline, we get to understand how it works together:

Step 1: Convert Grapheme (Text) to phoneme

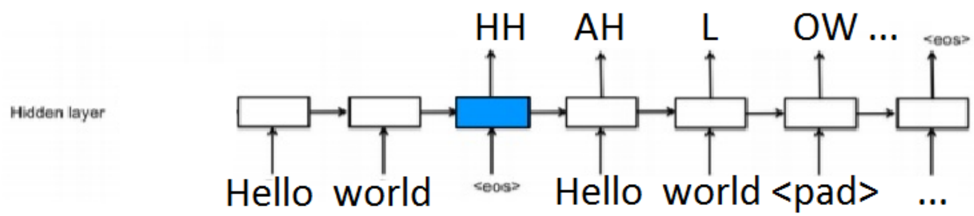
Step 2: Duration Prediction

Step 3: Fundamental Frequency Prediction

Deep Voice

Step 1: Convert Grapheme (Text) to phoneme

Source: Hello world
Target: HH AH L OW . W ER L D



Languages such as English are peculiar in that they aren’t phonetic. For instance, take the following words (adapted from here) that all use the suffix “ough”:

though (like o in go) - through (like oo in too) - cough (like off in offer)

Notice how they all have fairly different pronunciations even though they have the same spelling. Phonemes are the different units of sound that we make. Combining them together, we can recreate the pronunciation for almost any word. Examples:

Input - “It was early spring” - Output - [IH1, T, ., W, AA1, Z, ., ER1, L, IY0, ., S, P, R, IH1, NG, .]

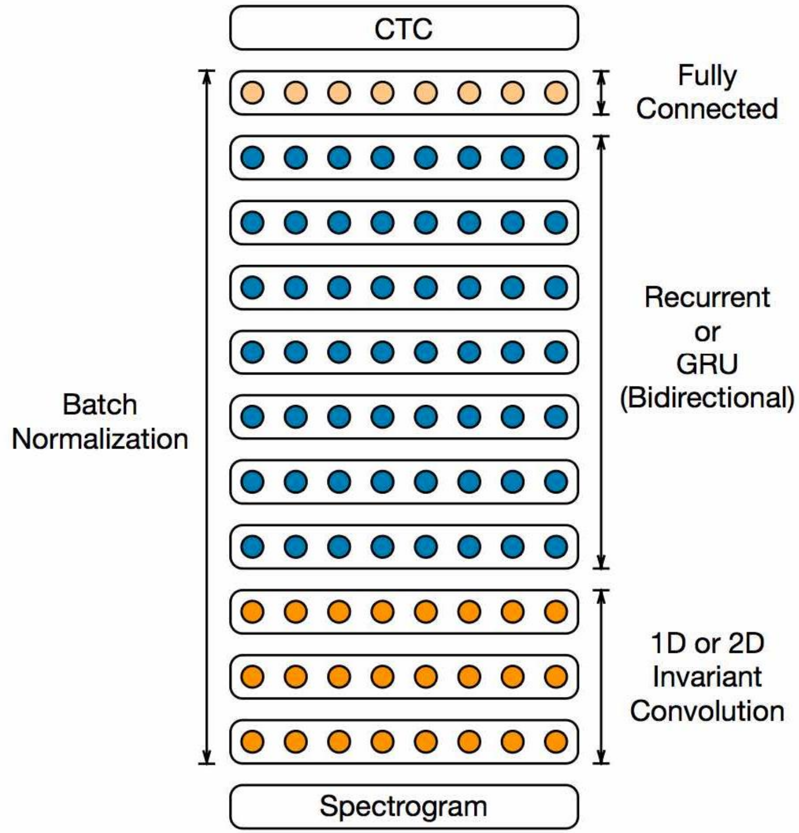
Deep Voice – The Inference Pipeline

Deep Voice use a single architecture to jointly predict phoneme duration and time-dependent fundamental frequency. The input to the model is a sequence of phonemes with stresses, with each phoneme and stress being encoded as a one-hot vector.

Step 2: Duration Prediction

Here's what will happen to our example sentence at this step:

Sentence: "It was early spring"
Input - [IH1, T, ., W, AA1, Z, ., ER1, L, IY0, ., S, P, R, IH1, NG, .]
Output - [IH1 (0.1s), T (0.05s), . (0.01s), …]



Deep Voice – The Inference Pipeline

Step 3: Fundamental Frequency Prediction

Deep Voice predict the tone and intonation of each phoneme to make it sound as human as possible.

It is especially important in languages like Mandarin where the same sound can have an entirely different meaning based on the tone and accent. Predicting the fundamental frequency of each phoneme helps us do just this.

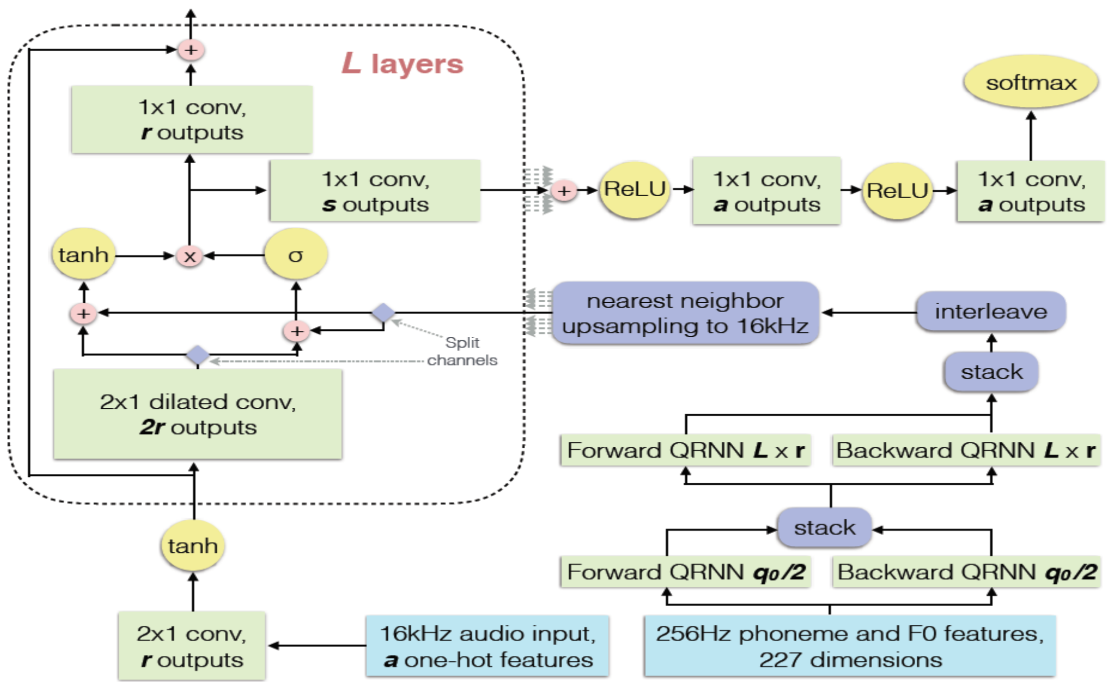
The frequency tells the system exactly what approximate pitch or tone the phoneme should be pronounced at.

Here’s what will happen to our example sentence at this step:
Input - [IH1, T, ., W, AA1, Z, ., ER1, L, IY0, ., S, P, R, IH1, NG, .]
Output - [IH1 (140hz), T (142hz), . (Not voiced), …]

Deep Voice – The Inference Pipeline

Step 4: Audio synthesis
The final step to creating speech is taking together the phonemes, the durations, and the frequencies to output sound. Deep Voice achieves this step using a modified version of DeepMind's WaveNet.

Baidu team modifies WaveNet by optimizing its implementation especially for high frequency inputs. As such, Where WaveNet required minutes to generate a second of new audio, Baidu's modified WaveNet can require as little as just a fraction of a second



The modified WaveNet architecture.

Deep Voice – Benefit

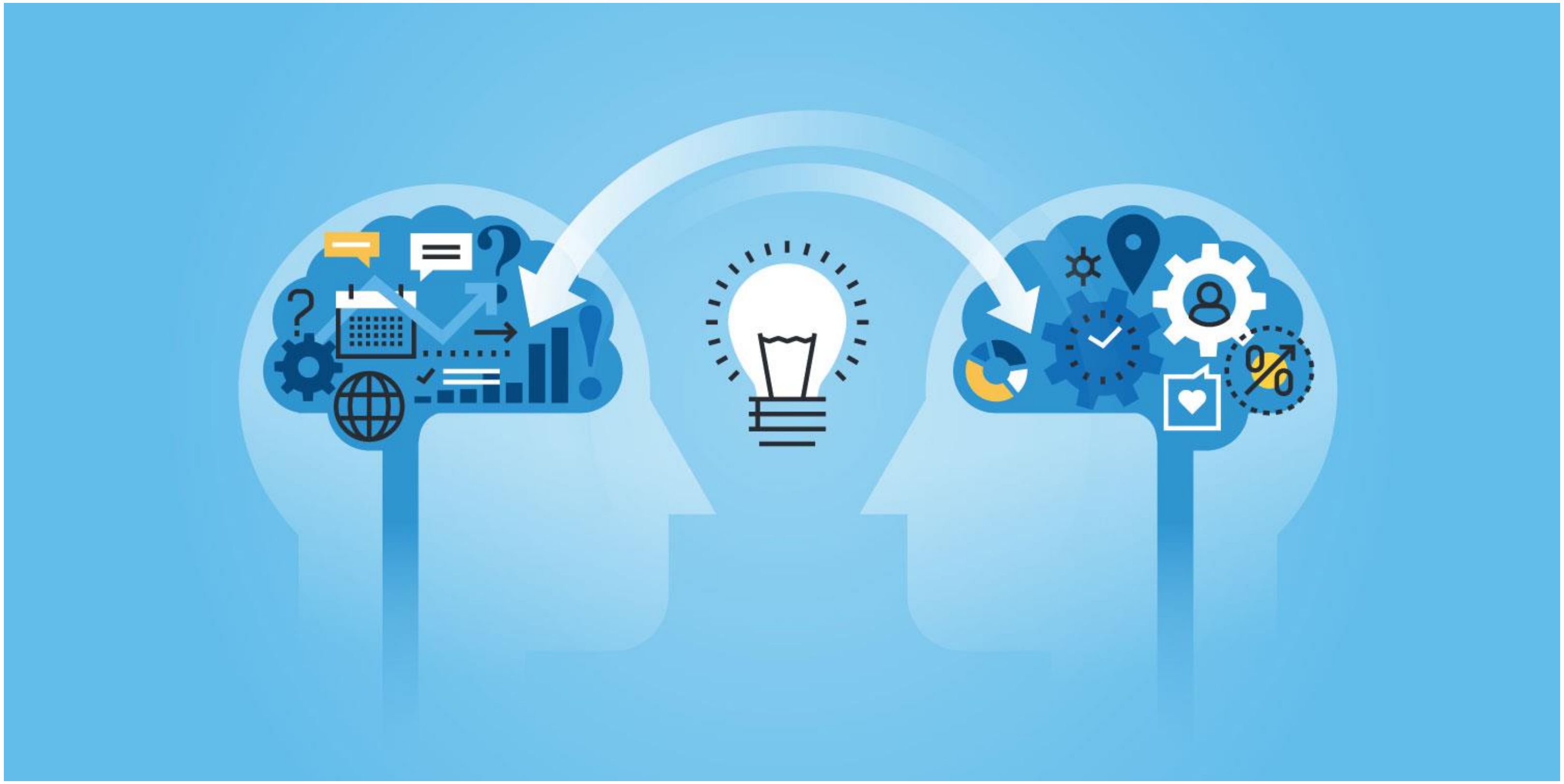
Benefit of TTS

The benefits of speech synthesis have been many, including computers that can read books to people, better hearing aids, more simultaneous telephone conversations on the same cable, talking machines for vocally impaired or deaf people and better aids for speech therapy.

NLP – More Details

Why is NLP Hard?

"At last, a computer that understands you like your mother"



Ambiguity

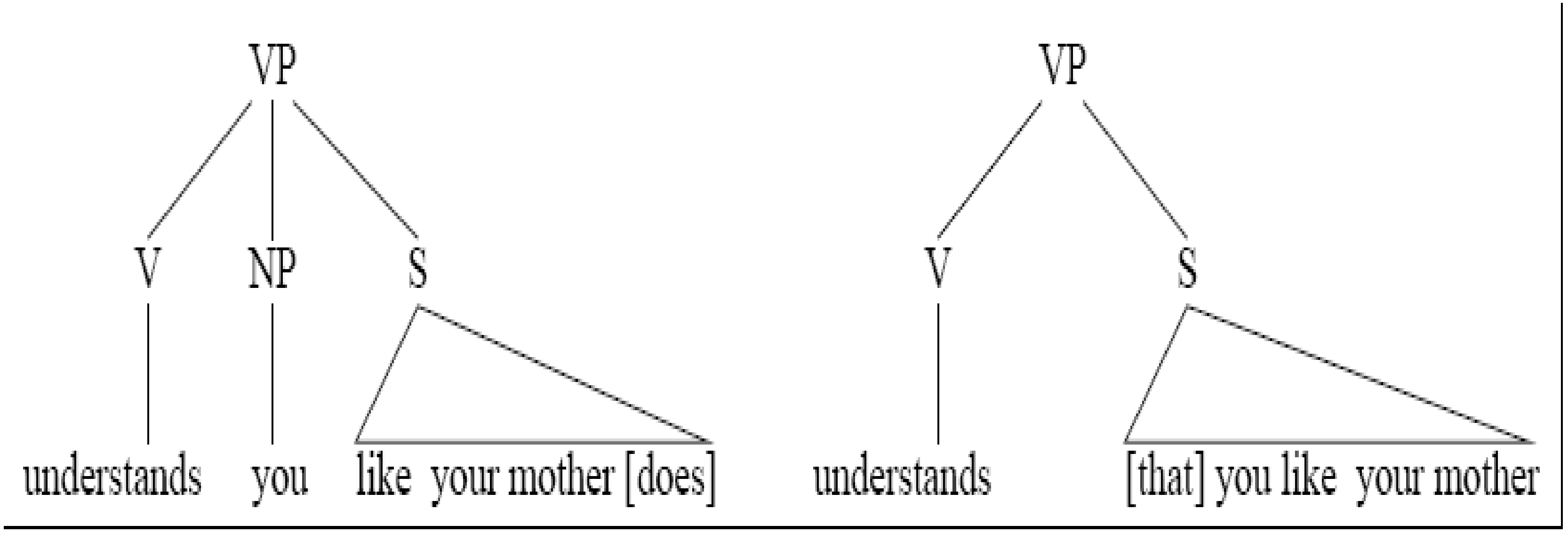
"At last, a computer that understands you like your mother"

1. (*) It understands you as well as your mother understands you?
2. It understands (that) you like your mother?
3. It understands you as well as it understands your mother?

Ambiguity at the Acoustic Level – Speech Recognition

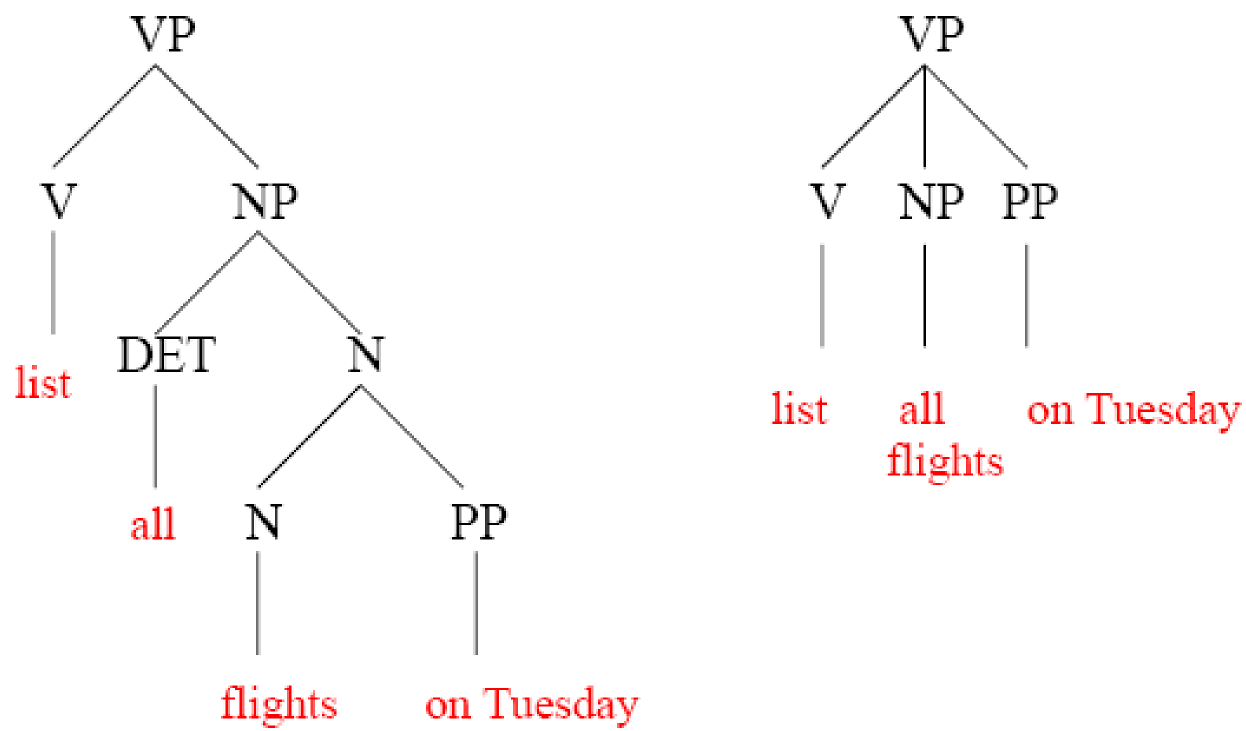
1. " ... a computer that understands you like your mother"
2. " ... a computer that understands you lie cured mother"

Ambiguity at the Syntactic Level



More Syntactic Ambiguity

Ambiguity at the Syntactic Level



Ambiguity at Many Levels

Ambiguity at the Semantic (meaning) Level

Two definitions of “mother”

- a woman who has given birth to a child
- a substance consisting of bacteria, used to produce vinegar (i.e., mother of vinegar)

This is an instance of word sense ambiguity

Ambiguity at the Discourse Level

- Alice says they’ve built a computer that understands you like your mother
- But she ...
- ... doesn’t know any details
- ... doesn’t understand me at all

Syntactic analysis

- Syntax can make explicit when there are several possible interpretations
  - *(Rice flies) like sand.*
  - *Rice (flies like sand).*
- Knowledge of ‘correct’ grammar can help finding the right interpretation
  - *Flying planes are dangerous.*
  - *Flying planes is dangerous.*

Syntax shows how words are related in a sentence

Visiting aunts ARE boring.

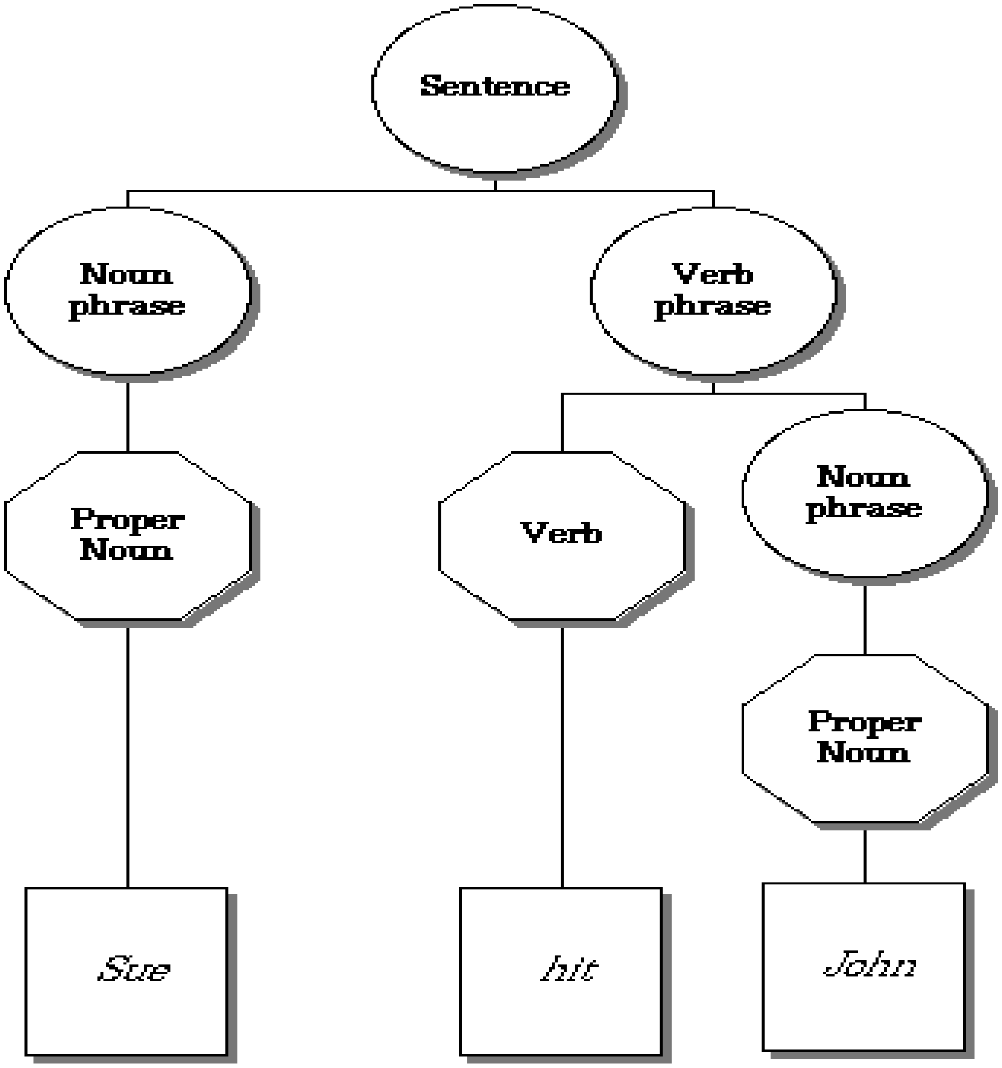
vs

Visiting aunts IS boring.

Subject verb agreement allows us to disambiguate here.

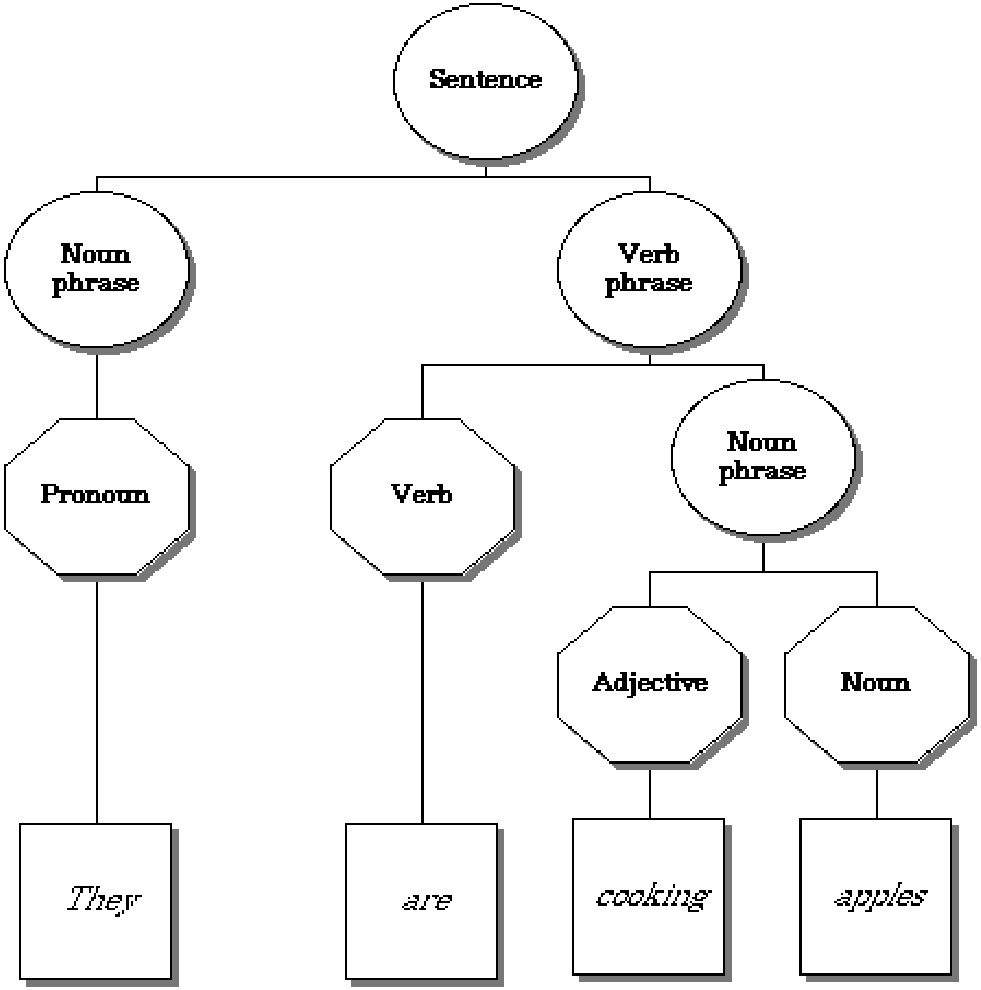
How do we represent syntax?

Parse Tree



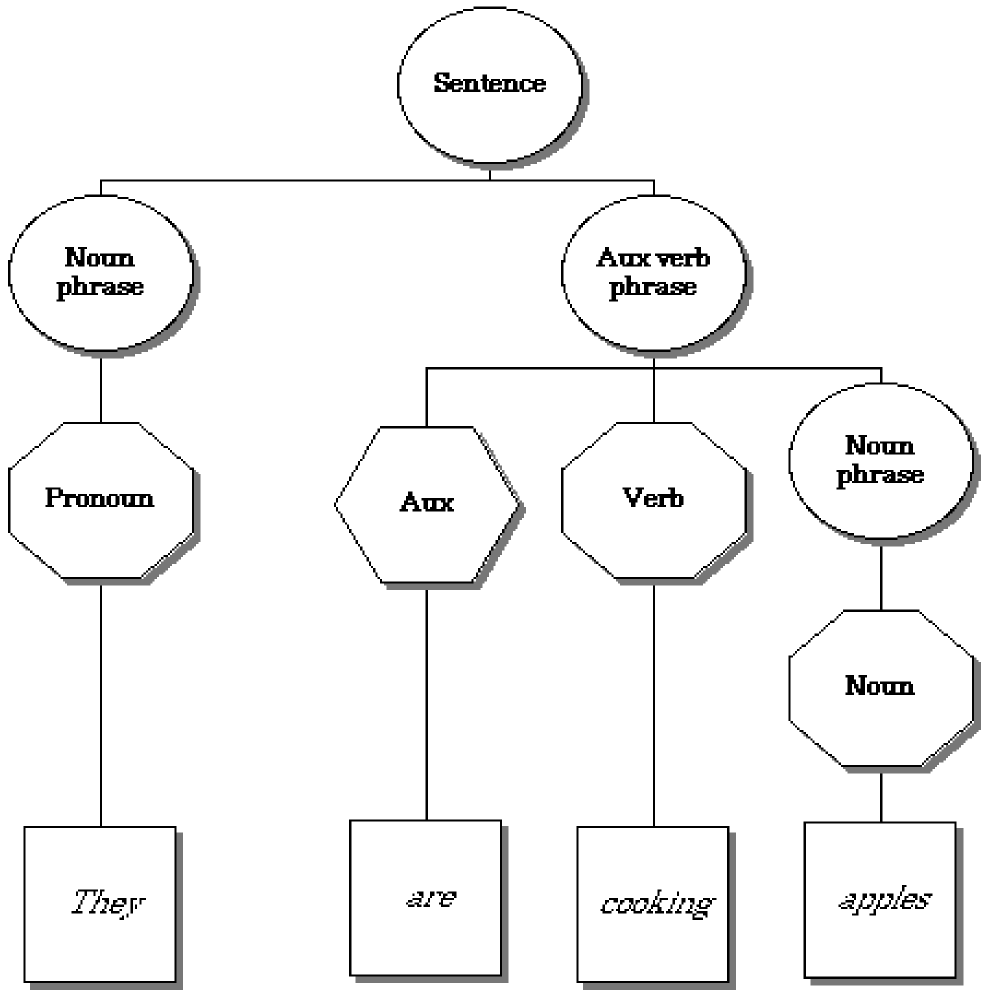
An Example

Parsing Sentence: “THEY ARE COOKING APPLES” – Parse 1



An Example

Parsing Sentence: “THEY ARE COOKING APPLES” – Parse 2



How do we represent syntax?

- **Using: LIST**

Sue hits John

[s, [np, [proper_noun, Sue]] ,

[vp, [v, hits],

[np, [proper_noun, John]]

Bottom Up vs. Top Down

Bottom - Up	Top - Down
John, hit, the, cat	s
prpn, hit, the, cat	s -> np, vp
prpn, v, the, cat	s -> prpn, vp
prpn, v, det, cat	s -> John, v, np
prpn, v, det, n	s -> John, hit, np
np, v, det, n	s -> John, hit, det,n
np, v, np	s -> John, hit, the,n
np, vp	s -> John, hit, the,cat
s	
• *Better if many alternative rules for a phrase* • *Worse if many alternative terminal symbols for each word*	• *Better if many alternative terminal symbols for each word* • *Worse if many alternative rules for a phrase*

Top down parsing (as a search procedure)

- $_1$ The $_2$ dog $_3$ cried

Step	Current state	Backup States	comment
1	((S) 1)		Initial position
2	((NP VP) 1)		Rule 1
3	((ART N VP) 1)		Rules 2 & 3
		((ART ADJ N VP) 1)	
4	((N VP) 2)		Match Art with **the**
		((ART ADJ N VP) 1)	
5	((VP) 3)		Match N with **dog**
		((ART ADJ N VP) 1)	
6	((V) 3)		Rules 4 & 5
		((V NP) 3)	
		((ART ADJ N VP) 1)	
7			Success

Chart Parsing

General Principles

A *Bottom-Up* parsing method

- Construct a parse starting from the input symbols
- Build constituents from sub-constituents
- When all constituents on the RHS of a rule are matched, create a constituent for the LHS of the rule

The *Chart* allows storing partial analyses, so that they can be shared.

Data structures used by the algorithm:

- **The Key:** the current constituent we are attempting to "match"
- **An Active Arc:** a grammar rule that has a partially matched RHS
- **The Agenda:** Keeps track of newly found unprocessed constituents
- **The Chart:** Records processed constituents (non-terminals) that span substrings of the input

Chart Parsing

Steps in the process

Input is processed left-to-right, one word at a time

1. Find all POS of the current word (terminal-level)
2. Initialize Agenda with all POS of the word
3. Pick a Key from the Agenda
4. Add all grammar rules that start with the Key as active arcs
5. Extend any existing active arcs with the Key
6. Add LHS constituents of newly completed rules to the Agenda
7. Add the Key to the Chart
8. If Agenda not empty – go to (3), else go to (1)

Chart Parsing Algorithm

Extending Active Arcs with a Key:

- Each **Active Arc** has the form:

  $<p_i>\ [A \rightarrow X_1 \ldots \bullet C \ldots X_m]\ <p_j>$

- A Key constituent has the form: $<p_j>C<p_k>$
- When processing the Key $<p_1>C<p_2>$, we search the active arc list for an arc $<p_0>[A \rightarrow X_1 \ldots \bullet C \ldots X_m]<p_1>$, and then create a new active arc $<p_0>[A \rightarrow X_1 \ldots C \bullet \ldots X_m]<p_2>$
- If the new active arc is a completed rule:
  $<p_0>[A \rightarrow X_1 \ldots C \bullet]<p_2>$, then we add $<p_0>A<p_2>$ to the Agenda
- After "using" the key to extend all relevant arcs, it is entered into the Chart

Chart Parsing - Example

The Grammar

(1) S $\rightarrow$ NP VP
(2) NP $\rightarrow$ ART ADJ N
(3) NP $\rightarrow$ ART N
(4) NP $\rightarrow$ ADJ N
(5) VP $\rightarrow$ AUX VP
(6) VP $\rightarrow$ V NP

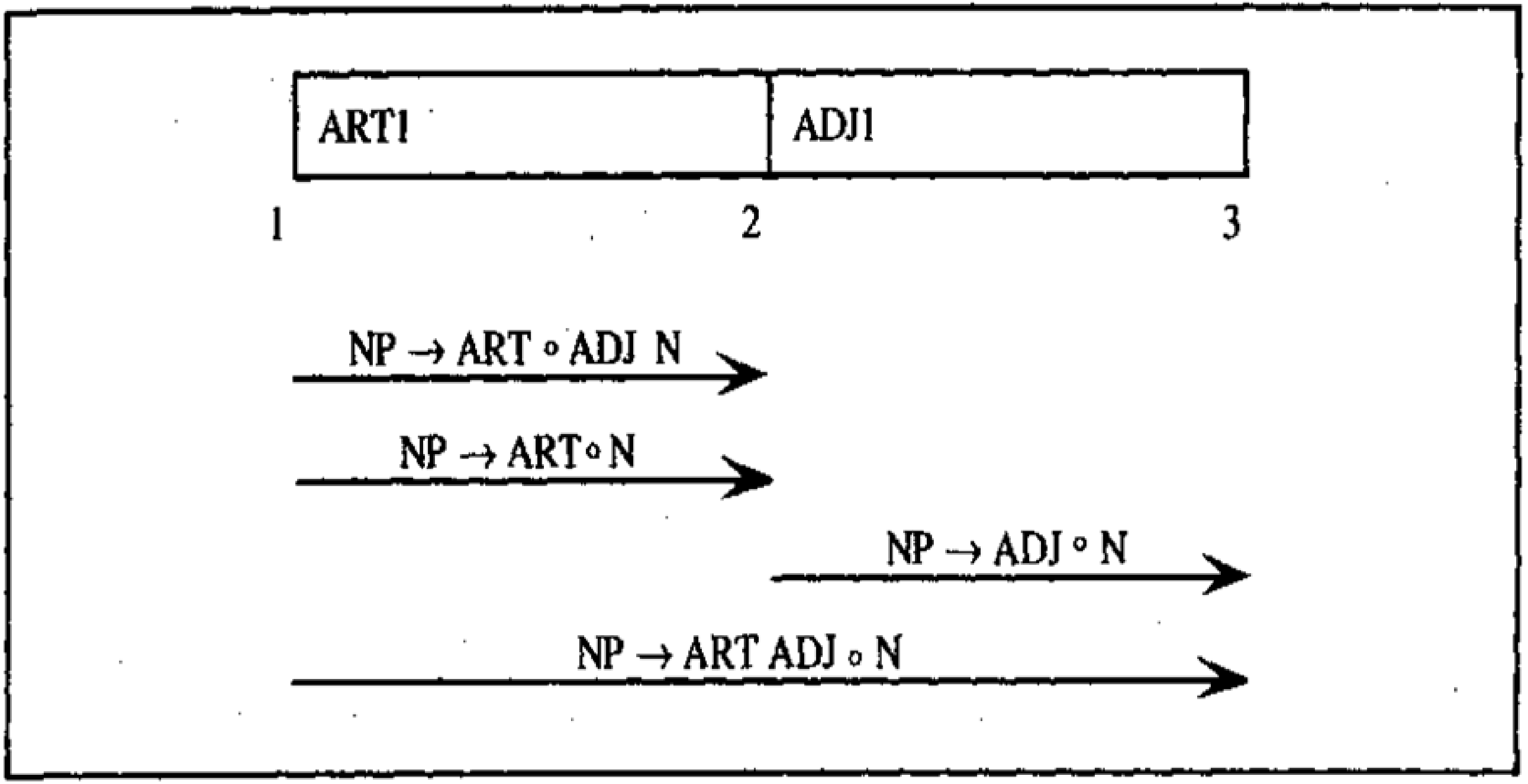
The Input

"x = The large can can hold the water"

POS of Input Words:

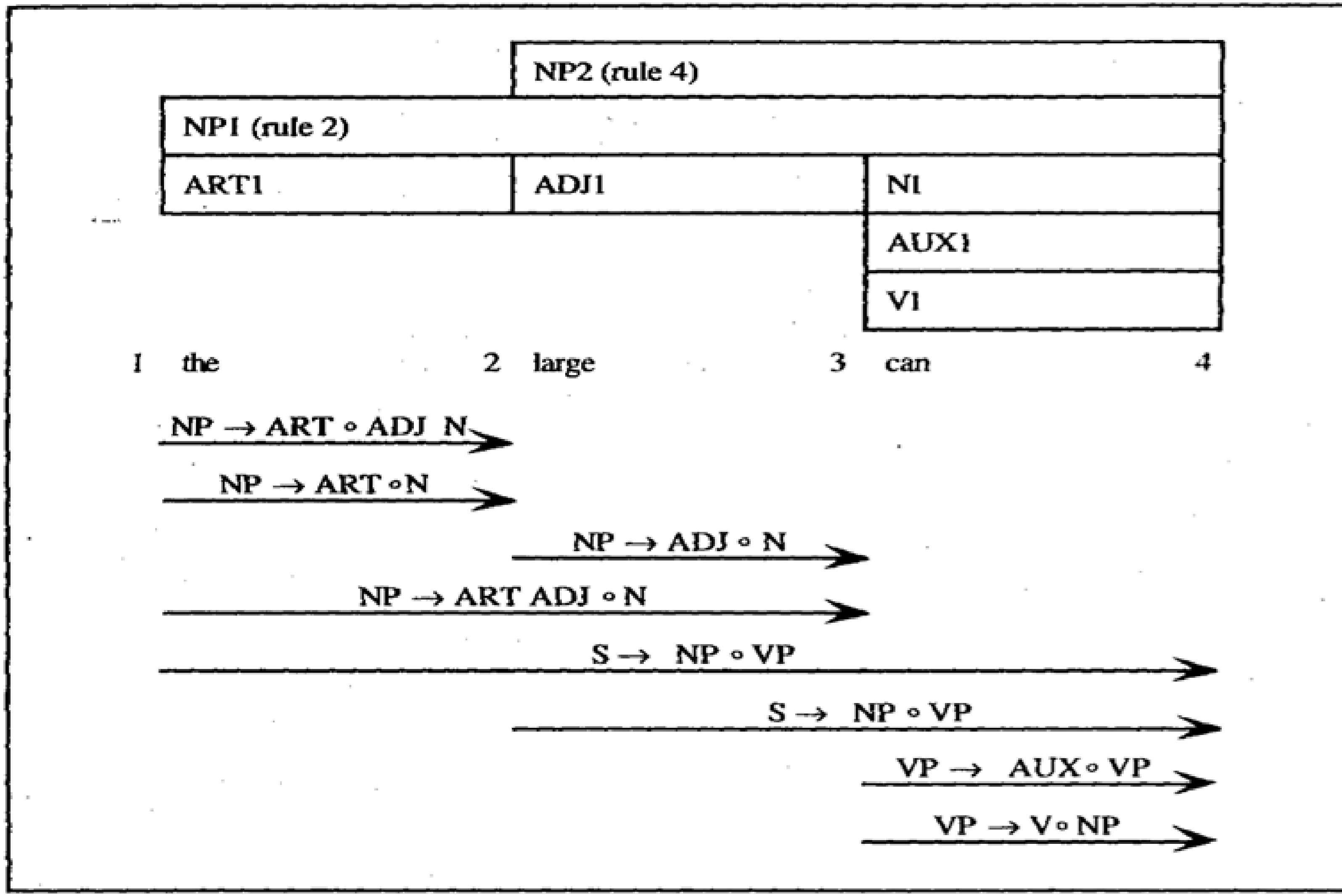
- the: *ART*
- large: *ADJ*
- can: *N, AUX, V*
- hold: *N, V*
- water: *N, V*

The large can can hold the water



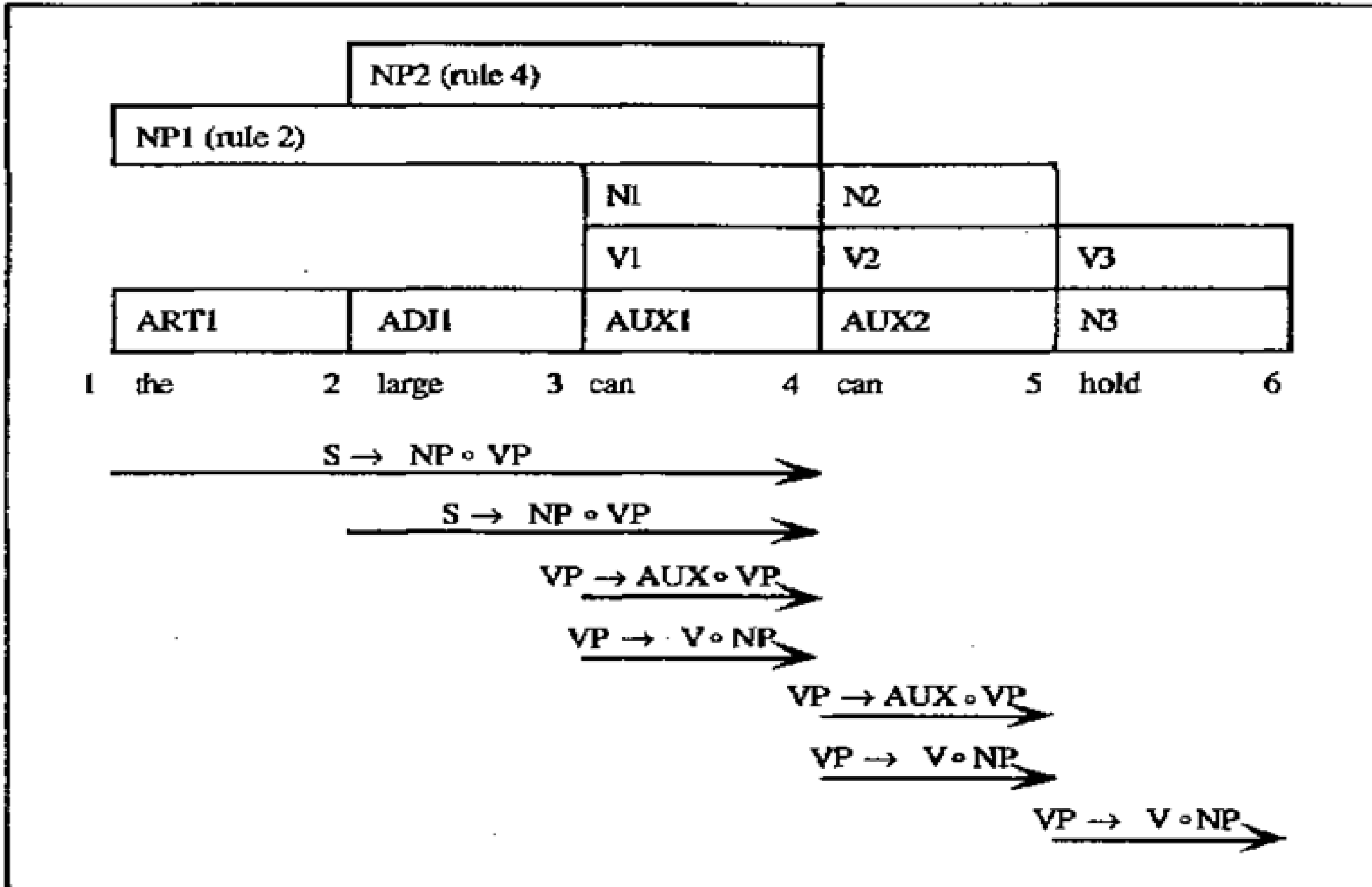
The chart after seeing an ADJ in position 2

The large can can hold the water



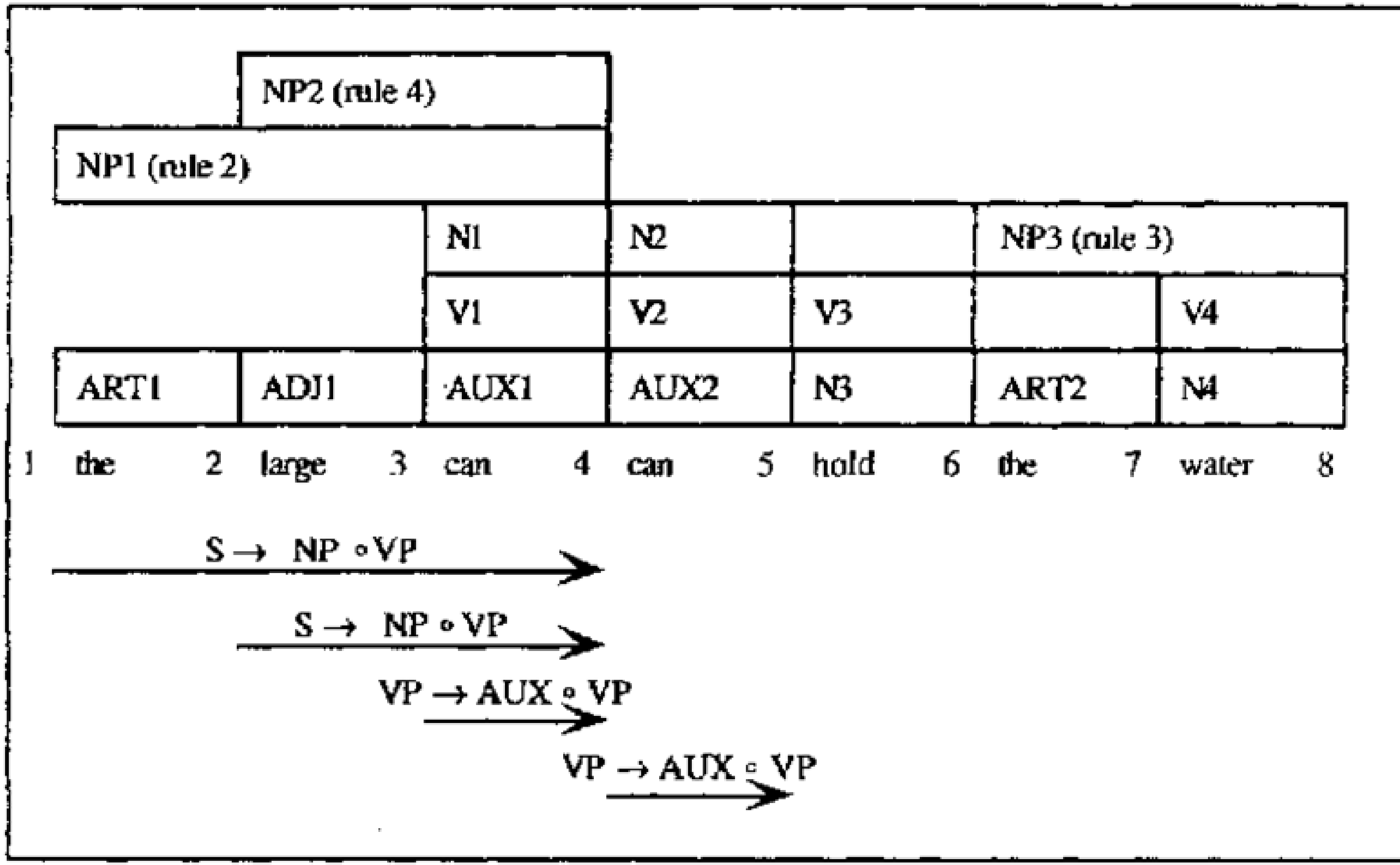
After parsing *the large can*

The large can can hold the water



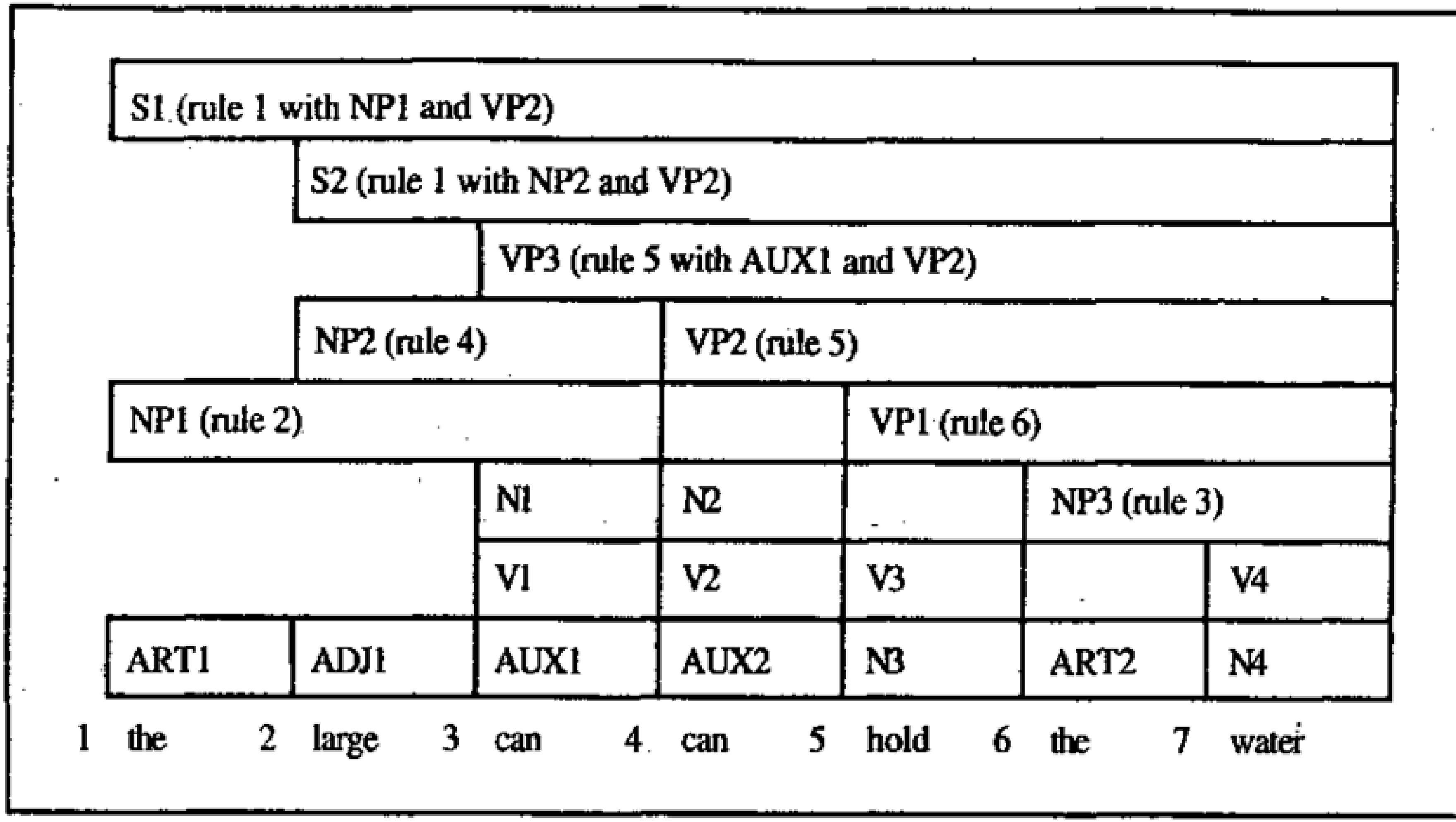
The chart after adding *hold*, omitting arcs generated for the first NP

The large can can hold the water



The chart after all the NPs are found, omitting all but the crucial active arcs

The final Chart



The final chart

Natural Language Processing – CASE STUDIES

Fpt Software | CAMPUS

News Monitoring System

A system to collect information about some world famous companies then classify them.
High quality product to scale system out to Big Data Environment at a reasonable efforts and costs.

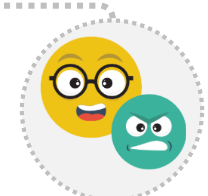
IT Support Email Chatbot

An automated email responding system.
Reduce cost of operation, improve customer satisfaction with accuracy of 91.5%.

Digital Advisor

Recognize customer's problem and provide instant answer.
Improve customer satisfaction while reducing cost of operation.

Sentiment Analysis



Identify, extract, quantify, and study affective states and subjective information.
Forecast market movement based on news, blogs and social media sentiment.

Voice Control System

Speech-to-text transformation is performed locally by client's mobile app.
Controlling TV becomes much easier and faster by using your own voice.

News Monitoring System

Business Overview

Customer is a British multinational corporation headquartered in London, United Kingdom, that provides risk management, insurance and reinsurance brokerage, investment banking, human resource solutions and outsourcing services. Aon has approximately 500 offices worldwide, serving 120 countries with 65,000 employees

Business Need

- The customer needs to develop a system to monitor information about some world famous companies.
  - Collect news articles that write about 12 world famous companies from 9 world popular English online News.
  - Classify news articles into 13 categories based on semantics of them.
  - Visualize results of Collector and Classifier.
- The system can be improved and scaled out by time.

Our Solution

- **Phase 1:** Build a proposal product for demo.
- **Phase 2:** Develop completely system for thousands companies and news sources. Deploy system on Amazon EMR (coming soon).

Benefit & Value

- The FPT Software team researched most effective method to classify online news article.
- The FPT Software team delivered a high quality proposal product and solution to scale system out to Big Data Environment at a reasonable efforts and costs.

News Monitoring System

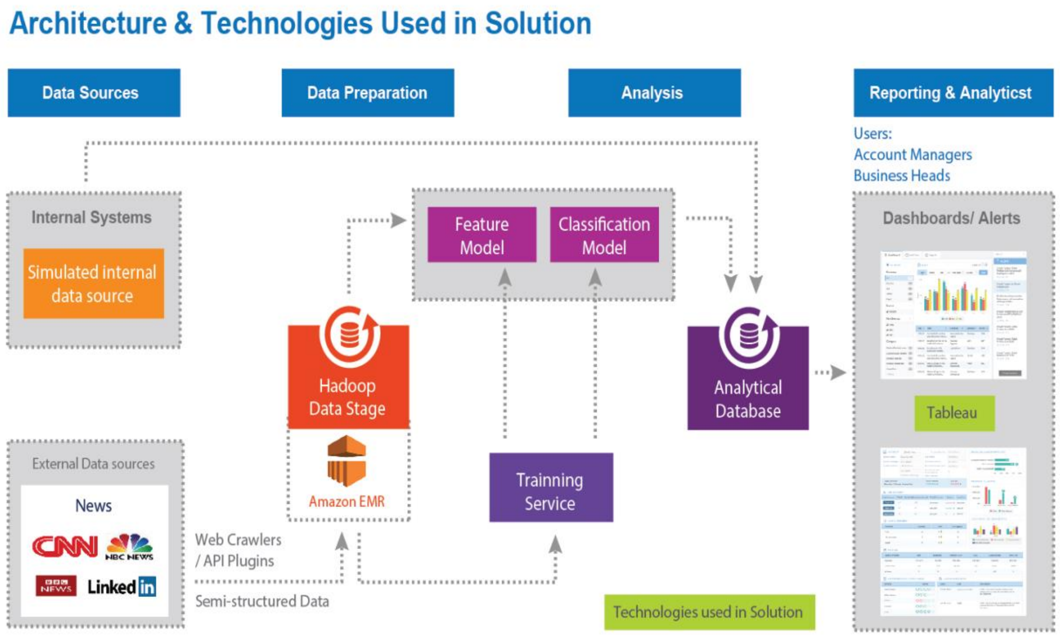
Technical Overview

Project Information

- **Team Size:** 4 senior researchers, 8 developers.
- **Duration:** 4 months of proposal and solution development.

Technologies Used

- **Development:** Java, Mysql, R,…
- **Tools:** LibSVM, Standford NLP Libs, R Shiny, Tableau,…
- **Research Methods**: Supervised Learning, Data Sampling, Voting, Word Vector, Ontology,…



Technical Details

- **Phase 1** – Proposal Product Development:
  - Develop a Crawler Program to collect Online News Articles which write about 12 concerned companies.
  - Develop a Classifier Program to classify Online News Article in 13 categories.
  - Develop a Visualization Interface that display News Articles and allow end user interact with interface to show data as their way.
  - Develop a Correction Interface that allow Admin User improve performance classification.
- **Phase 2:** Provide most effective solution to scale out system and deploy to Amazon EMR.

IT Support Email Chatbot

Business Overview

The customer is Singapore's fully-integrated info-communications company, offering a full range of information, communications and entertainment services for both consumer and corporate markets. entities.

Business Need

- The customer aims to build up an automated email responding system. They require to protect sensitive data coming from internal departments so the whole system must be deployed on premise.
- Main functions are:
  - Interaction with the Helpdesk chat bot to resolve problems related to account passwords/lock/unlock.
  - Sending an email to automatically create the ticket to solve these problems and view these tickets' status in web browser.
  - If the problem is out of scope or users get trouble with with the Email-/chat bot system, they will be connected to the real helpdesk agent and for support.

Our Solution

- FPT propose to use FPT Conversational Platform (FCP) which is FPT's intelligent service. This allows to add a natural language interface to an application to automate the interaction with end users.

Benefit & Value

- Improve customer satisfaction with current accuracy (based on similar evaluation methods as Intent Classification) of 91.5%.
- Free up human resource in Helpdesk department.
- Reduce cost of operation.

IT Support Email Chatbot

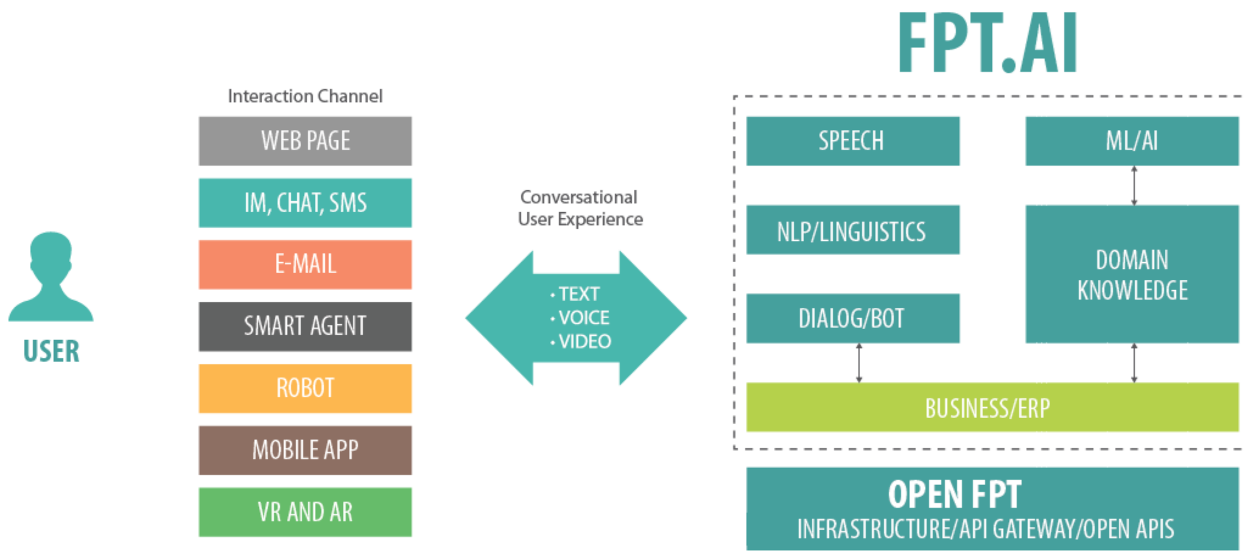
Technical Overview

Project Information

- **Team Size:** 6 people
- **Duration:** 1 month

Technologies Used

- Angular JS 2 Framework
- NodeJS v6.9.2 and Microsoft SQL Server 2012
- Microsoft Bot framework



Technical Details

- The system will have 3 main parts as below:
  - Ticket Dashboard: in which end-user can check their ticket status and email loop.
  - Back-end system: will have Ticket System and Bot System.
  - FPT.AI System.
- FPT.AI system includes:
  - Virtual agents and chat bots that can integrate and communicate on any channel or device.
  - Voice-enable IoT devices; users just talk to the machine to get information, control smart home products, get news, etc.

Digital Advisor

Business Overview

The customer is a company working in insurance and financial industry. Its products are life insurance, accident and health insurance, savings plans, employee benefits, credit life and pension services.

Business Need

- Ability to extract information and validate customer by comparing customer's input with data from database.
- Ability to recognize customer's problem and provide near instant answer based on a set of pre-defined answers, included:
  - Handle payment issues: provide account information, query payment by month, change credit card info, validate if a credit card number is supported, answer modes of payment, guide to set up payment.
  - Account troubleshooting: reset username/password, unlock account.
  - Locate beneficiary form: validate insurance certificate number and find the appropriate beneficiary form, provide user the link to download, send the form directly to any email address of user's choice.

Our Solution

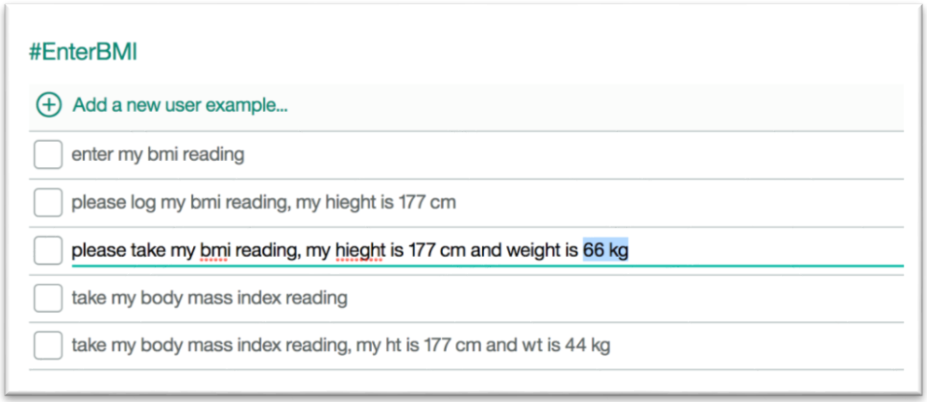
- IBM Watson Services: A natural language processing engine, using machine learning techniques.
- Server Side: written using Express Framework in NodeJS.

Benefit & Value

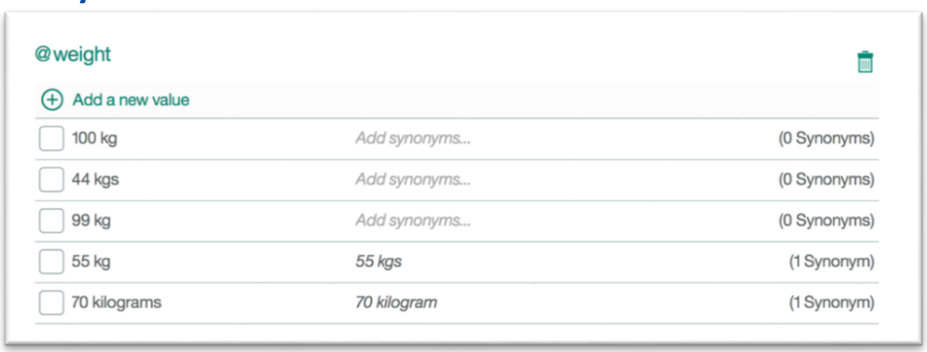
- Improve customer satisfaction while reducing cost of operation.

Sample screenshots

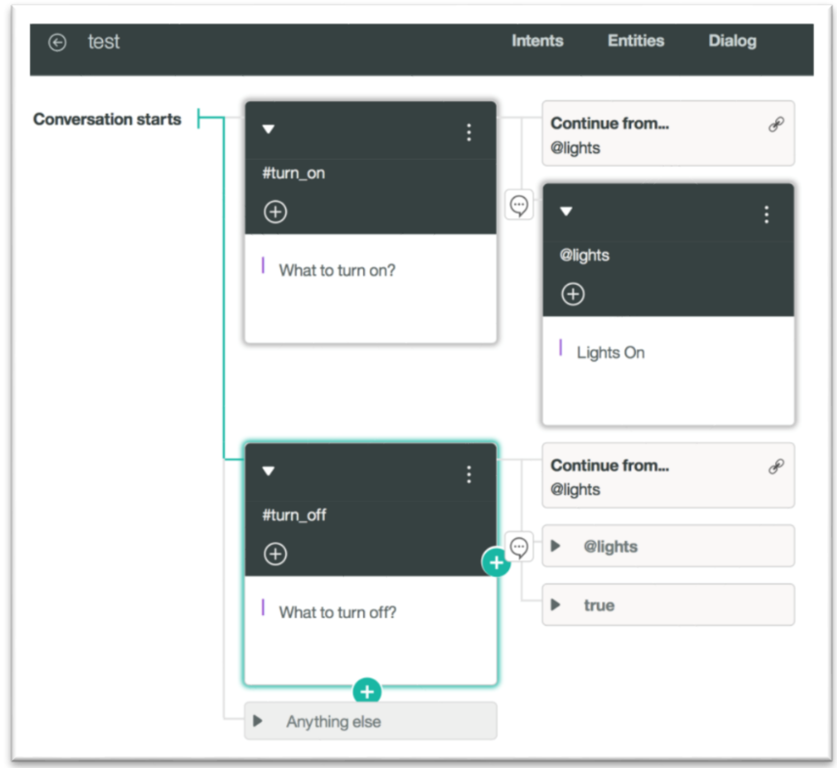
Intent



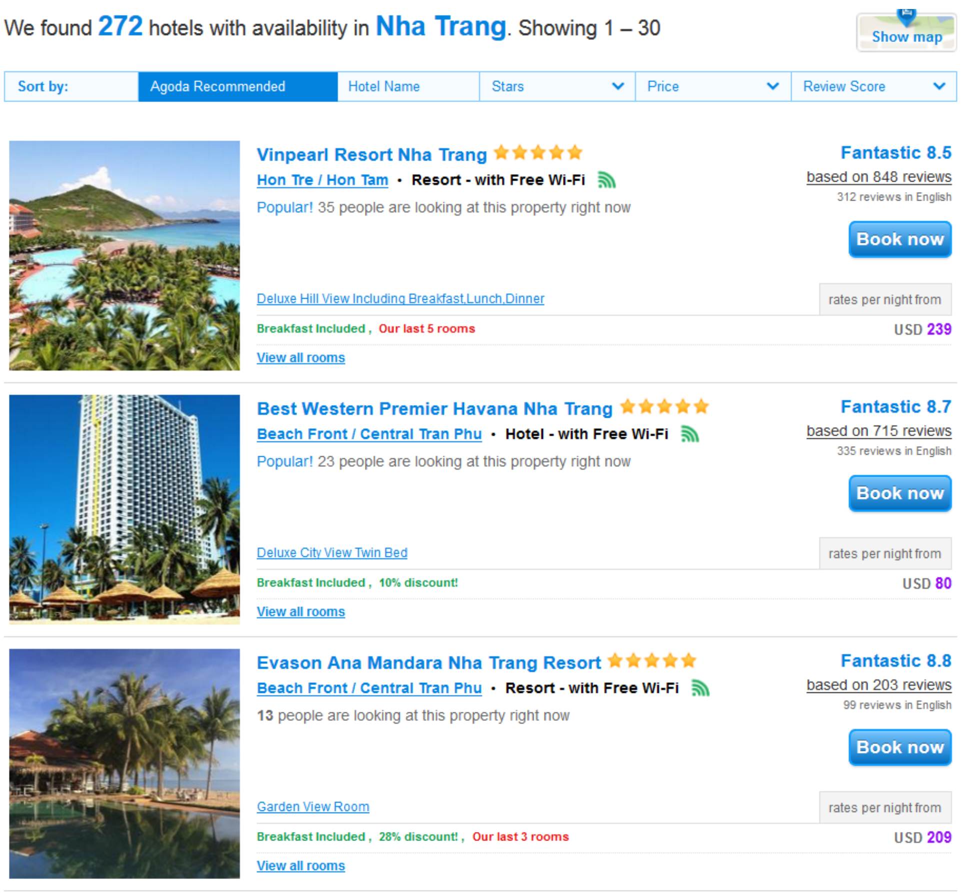
Entity



Context/Dialog/Conversation



Business Overview



How can I find a good hotel?

- Price?
- Location?
- Fasility?
- Service?
- Food?

It was a wonderful trip.

That hotel provides very bad services.

Sentiment Analysis

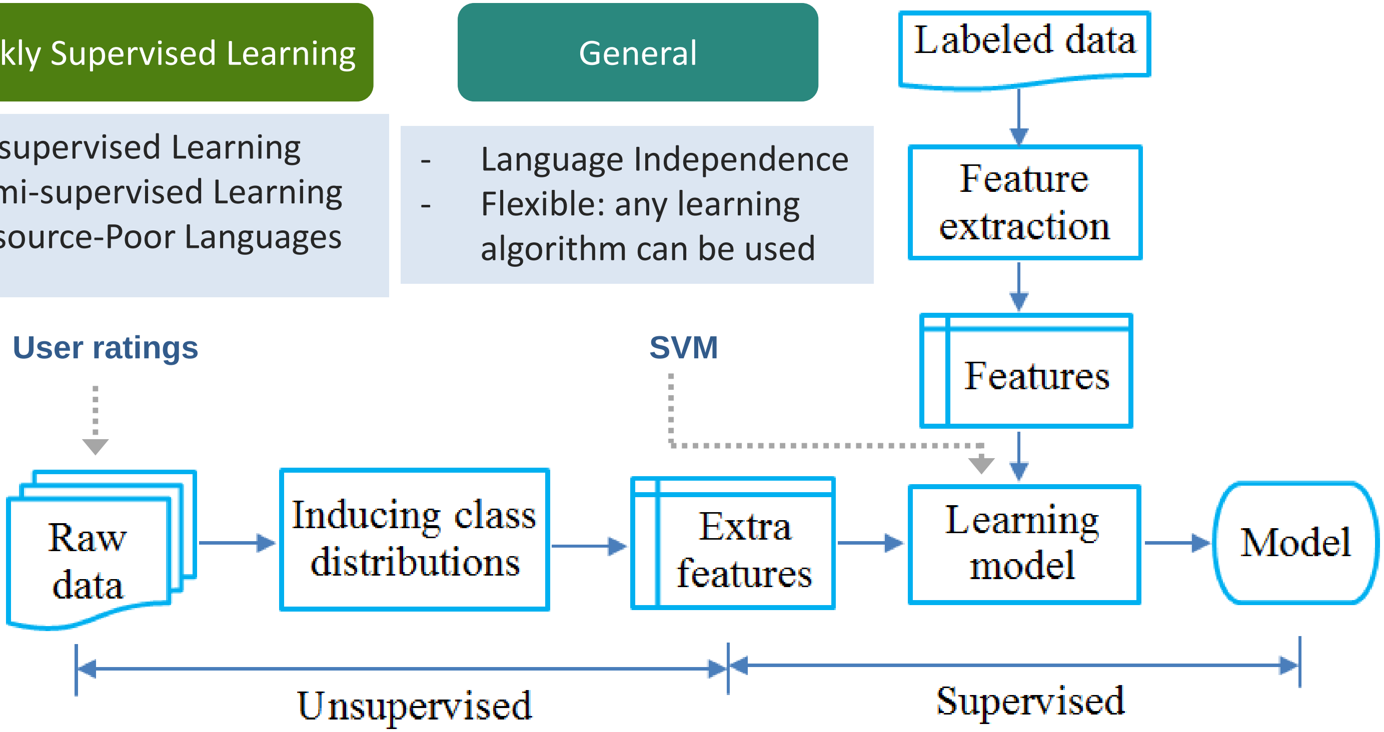
Our Solution

Weakly Supervised Learning

- Unsupervised Learning
- Semi-supervised Learning
- Resource-Poor Languages

General

- Language Independence
- Flexible: any learning algorithm can be used

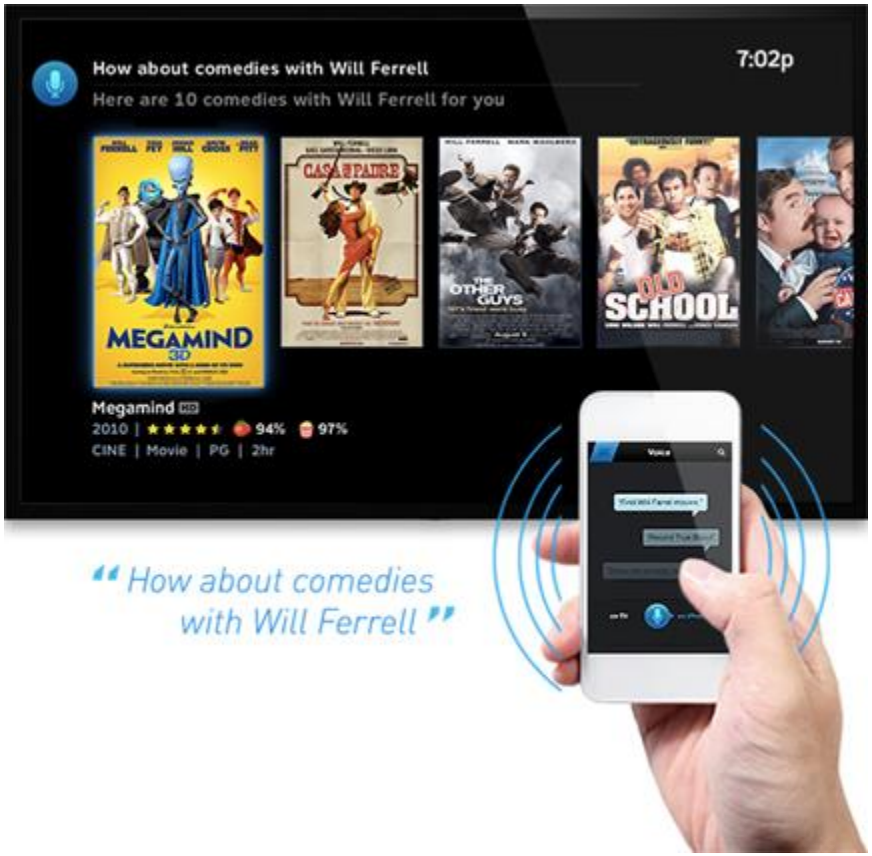


Benefit & Value

- Improve customer experience
- Forecast market movement based on news, blogs and social media sentiment
- Identify the clients with negative sentiment in social media or news and to increase the margin for transactions with them for default protection

Voice Control System

Business Overview



Business Need

- The customer is the America's #1 satellite TV with almost 30 million subscribers and hundreds of full-time HD channels.
- The customer wants to provide the best experience for their subscribers with an innovative, efficient and easy-to-use user interaction.

Our Solution

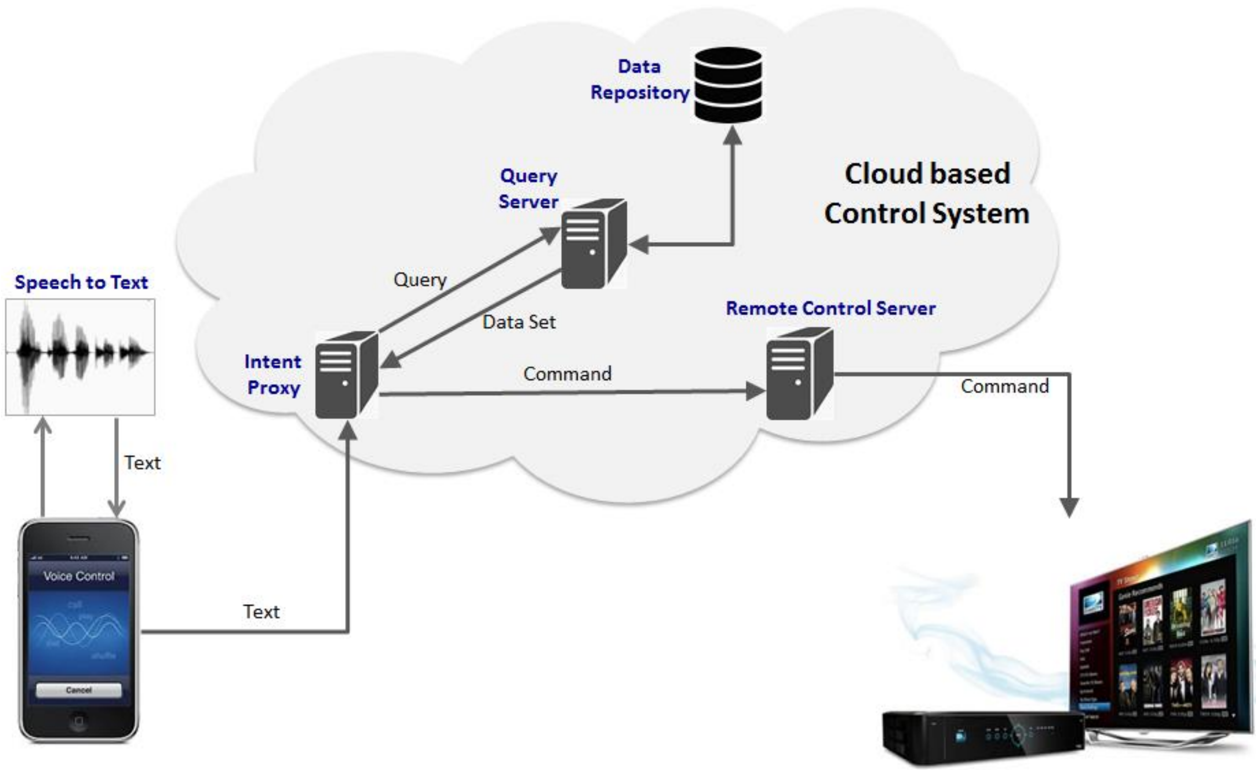
- We need to transform voice commands into text then find the best matched instruction.
- Speech-to-text transformation is performed locally by client's mobile app. Instruction mapping is processed on cloud.
- The system works with English speakers first. Another languages will be supported later.

Benefit & Value

- Controlling TV becomes much easier and faster by using your own voice.

Voice Control System

Technical Overview



Project Information

- **Team Size:** 28.
- **Duration:** 12 months of research & developments

Technical Details

- **Development:** Java.
- **Tools:** Eclipse.
- **Research Methods**: Clustering, Context Refinement Detection, Semantic Graph, Conditional Random Fields…

THANK YOU!