Microsoft

清華大學 Tsinghua University

从0到1搭建对话机器人

李佳芮

Founder & CEO of JuziBot
Wechaty 联合作者
Microsoft AI MVP
著有《Chatbot 从0到1》

MVP
Microsoft®
Most Valuable Professional

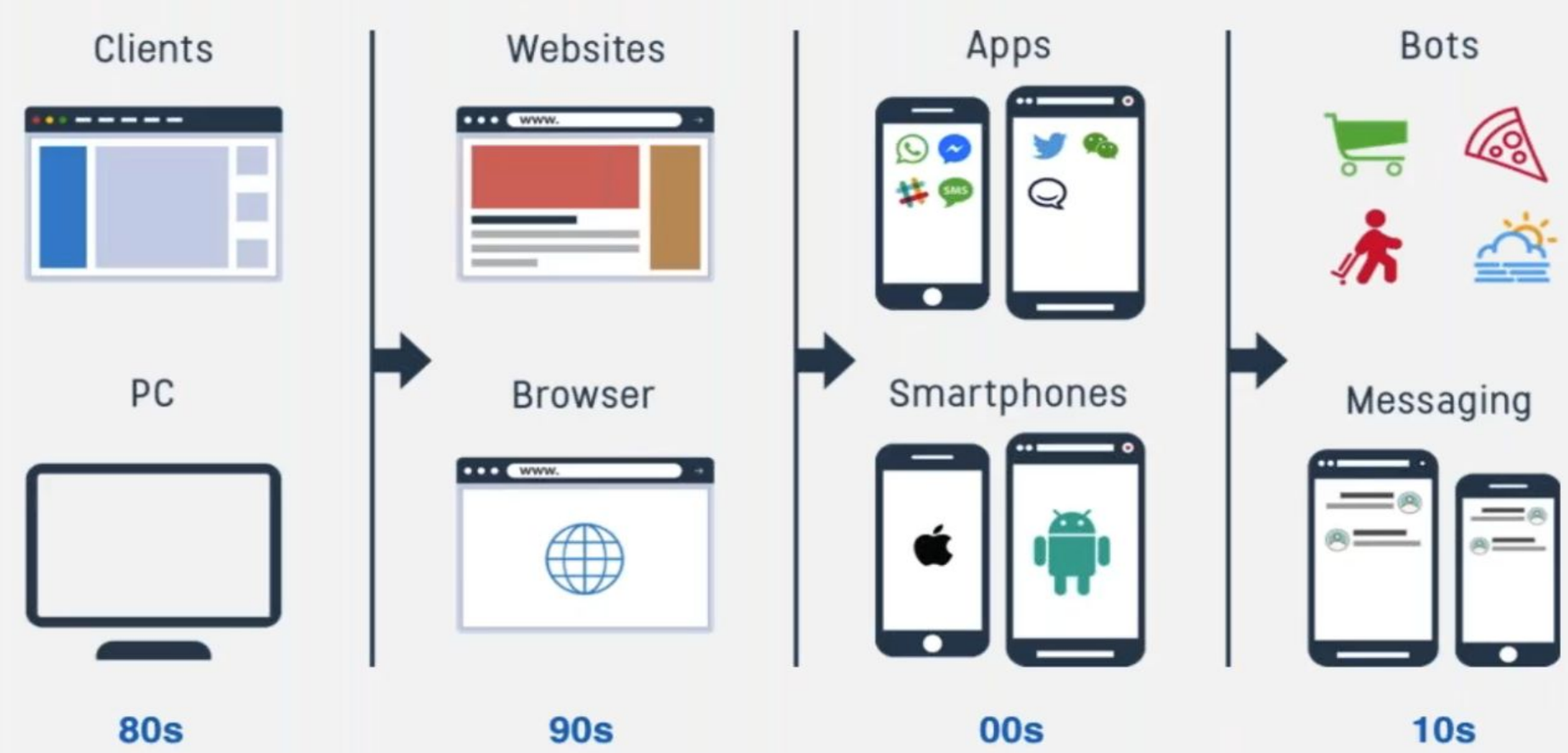
李佳芮 Rui Li

李佳芮，句子互动创始人，连续创业者，微软AI MVP，GitHub 7000+ Stars开源项目Wechaty联合作者，创建并管理了覆盖全球的微信聊天机器人开发者社区，《Chatbot从0到1》作者。句子互动围绕微信生态为客户提供智能营销和销售服务，帮助企业引流并实现转化，客户覆盖教育、保险、大健康等多个领域。曾入选百度AI加速器 和 Facebook 大陆首期加速器，获得PreAngel、Plug and Play，Y Combination, TSVC和阿尔法公社多家中美机构投资。

对话机器人介绍

对话系统趋势分析

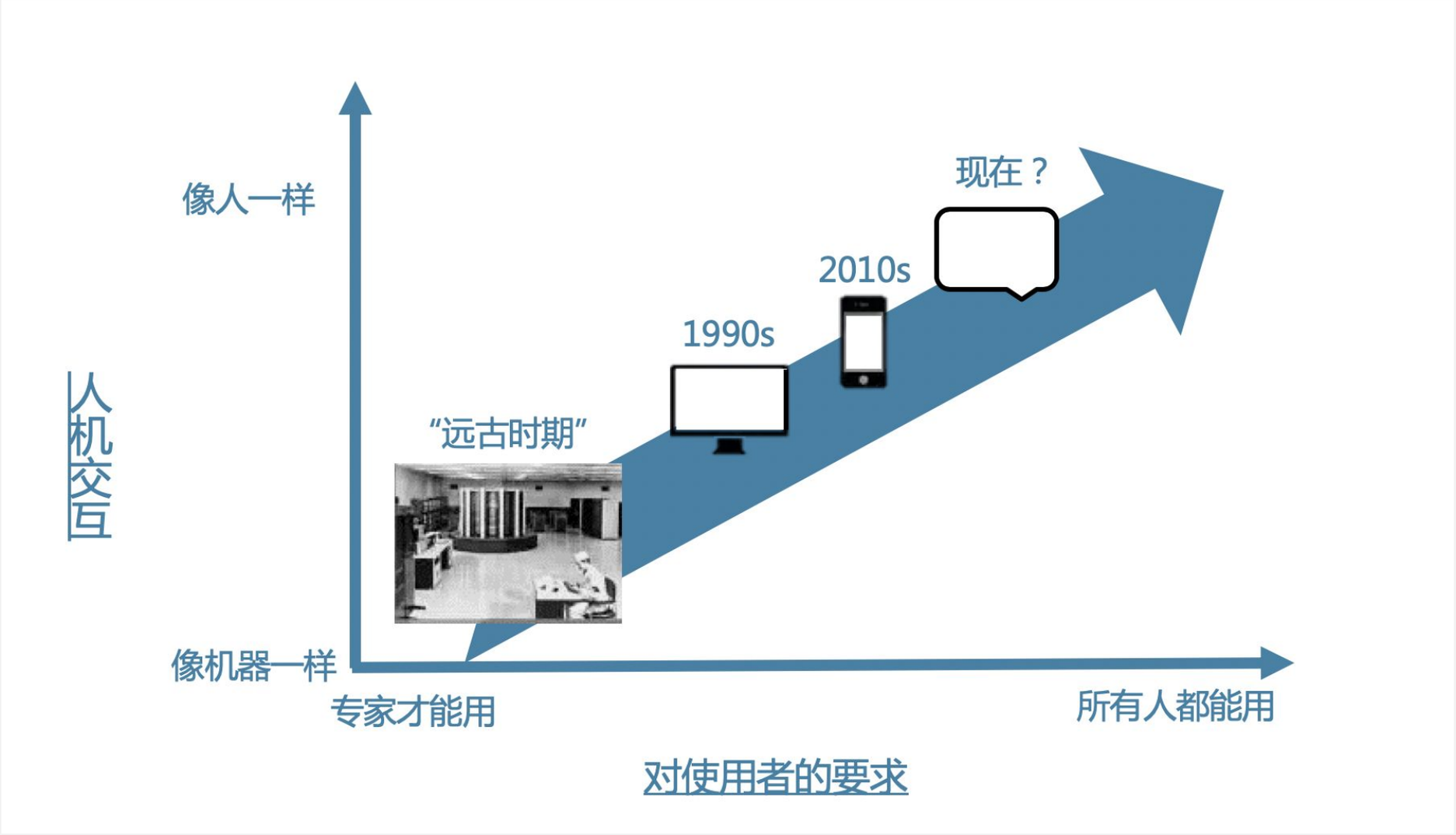
人类一直在寻找更加便捷的人机交互方式



信息的交互方式一直在演进，从PC、到互联网，再到移动互联网，交互方式越来越便捷。

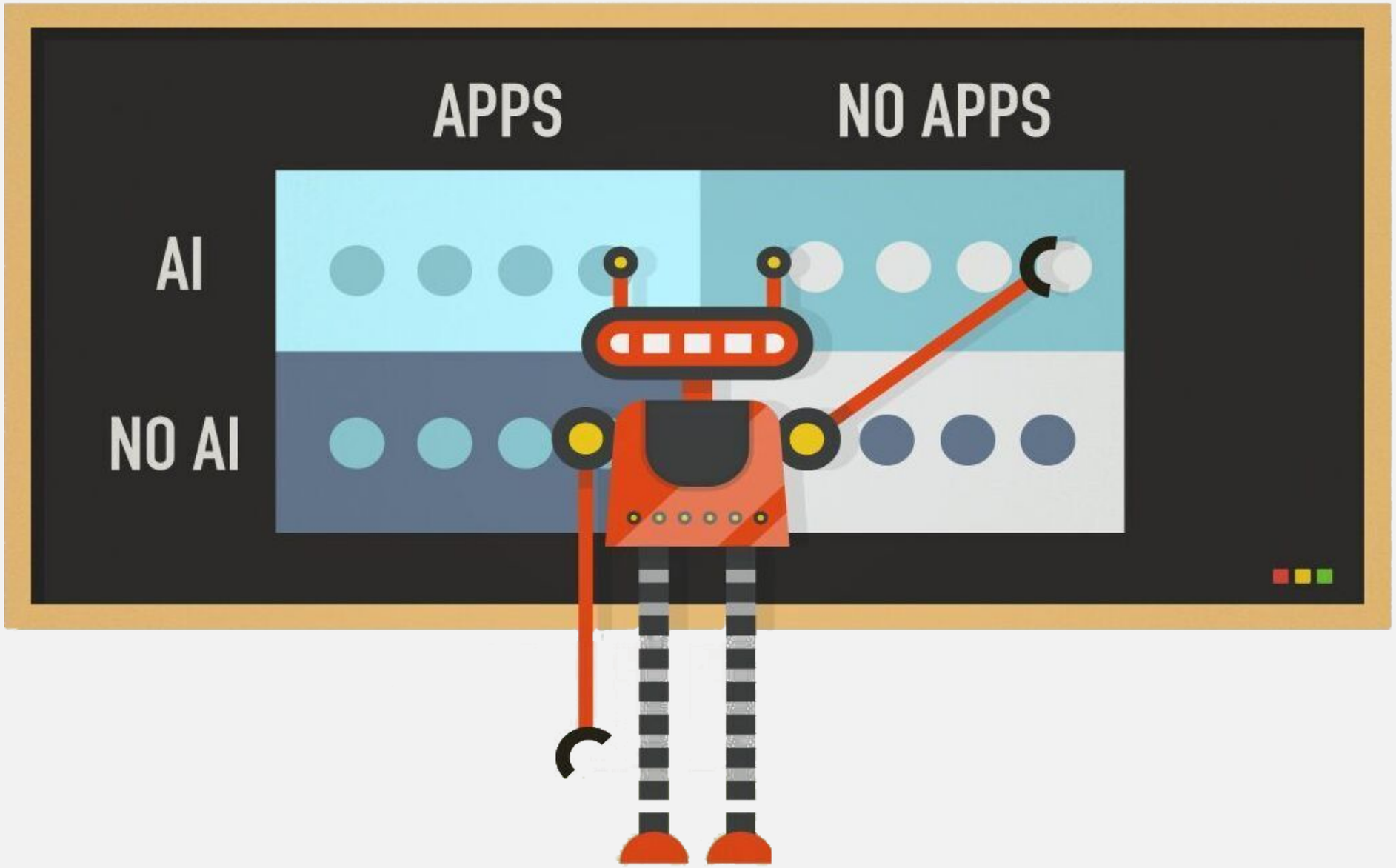
Source: https://www.gupshup.io/developer/home

交互的趋势

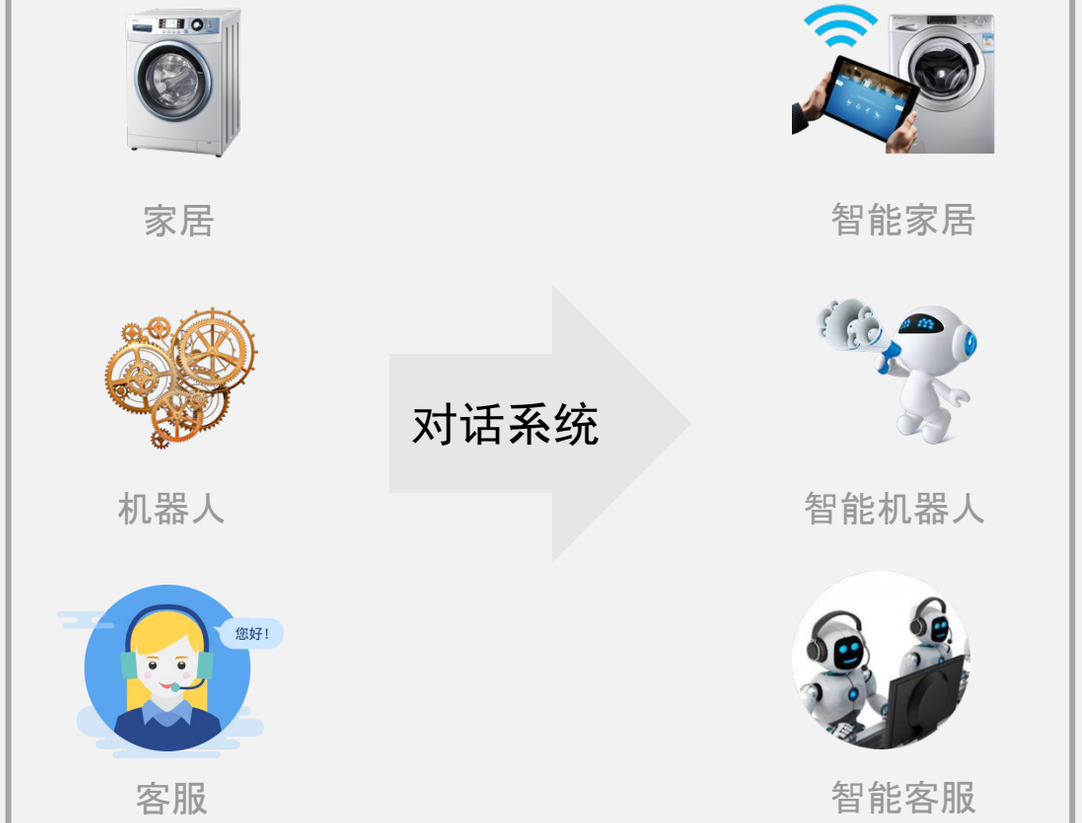


Source: S先生

交互的趋势

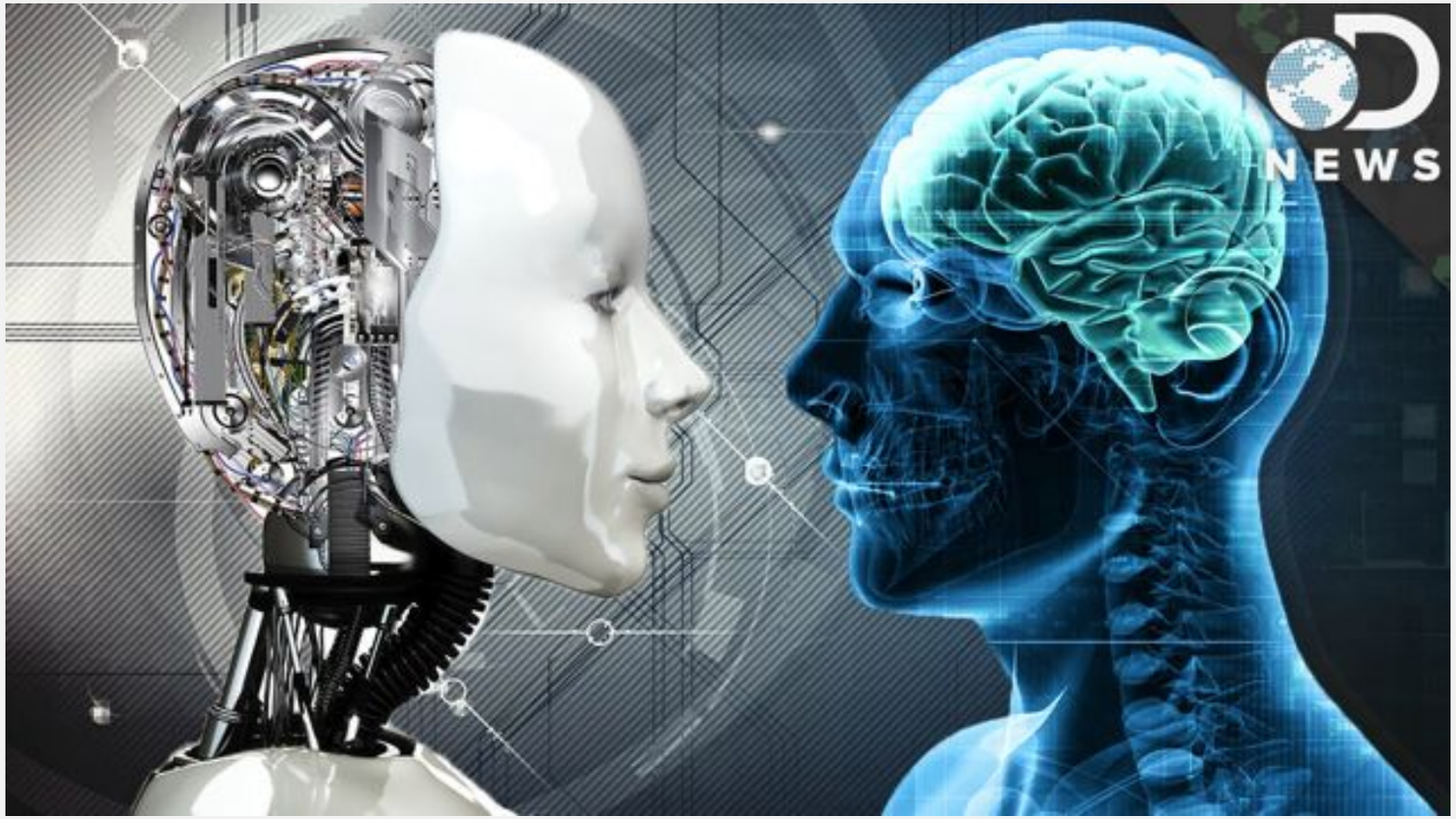


对话交互是下一个大趋势



系统将是人工智能时代的必要组件，任何产品，依赖对话系统为之赋予智能的能力

对话交互是一种新的UI



网站和APP 强制用户像机器人一样去思考问题

对话系统 强制机器像人一样思考问题

APP VS BOT

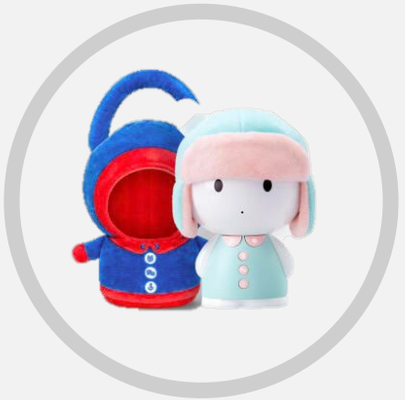
❌ App fatigue- Users don’t like download
❌ User acquisition costs are high
❌ Only short-tail apps popular
❌ High cost of development & upgrade & marketing

√ No installation
√ Works even in low-network areas
√ Zero UI
√ Low cost of development and upgrade

对话交互产品形态

商用机器人

家用机器人



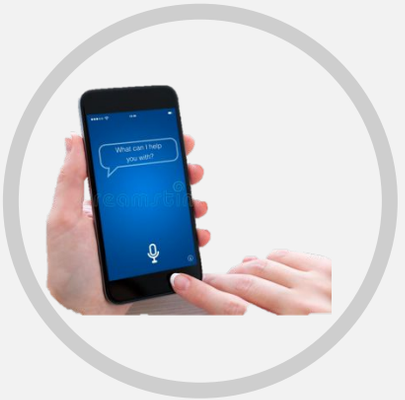
儿童故事机

智能音箱

智能家居

车载系统

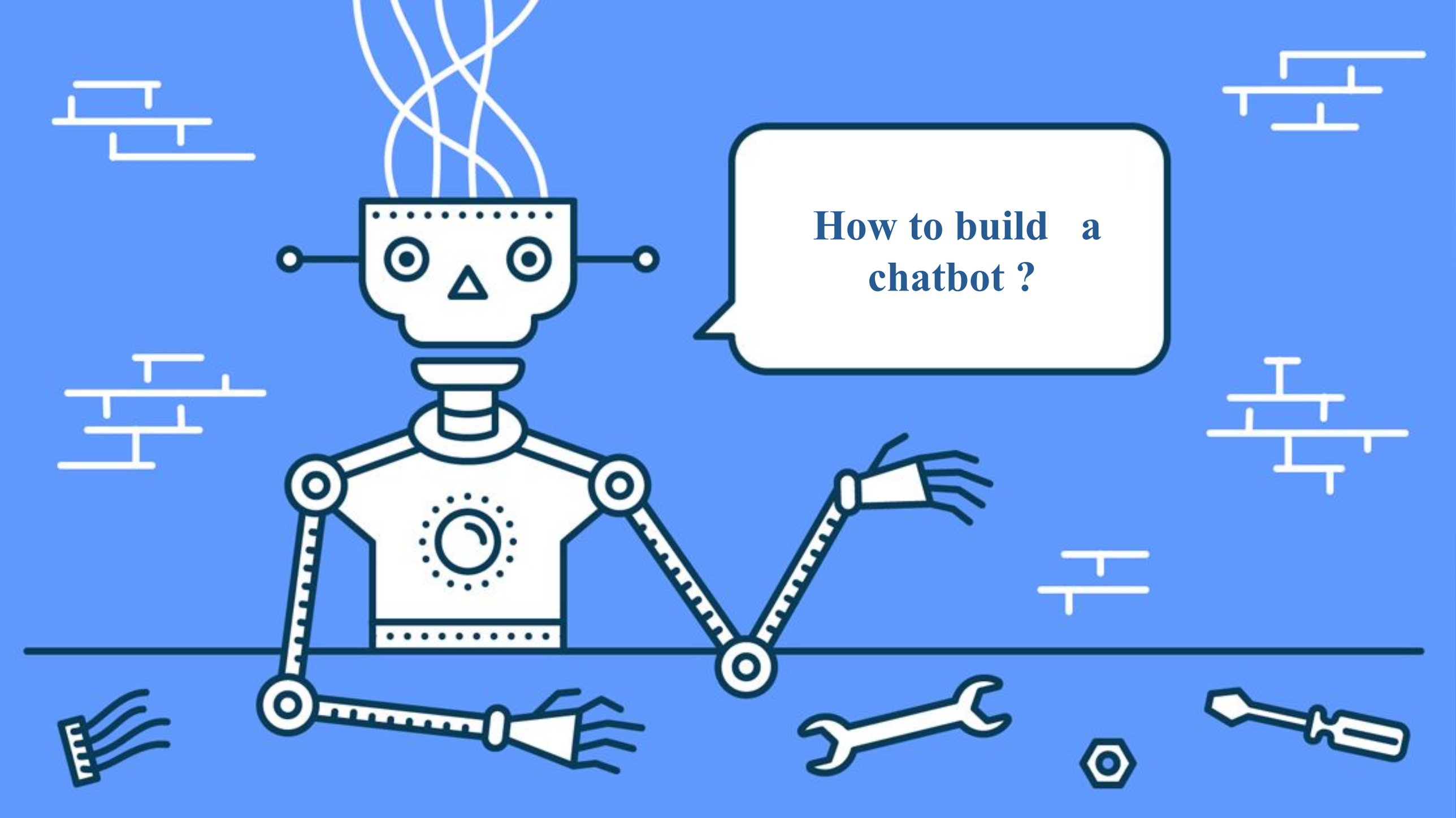
智能客服



个人助手类产品

对话系统的分类

分类	解决问题领域	平台系统开放性	技术方案	优化目标
任务型	有任务目标，且需要参数化请求	一家公司开发，直接面向C端消费者，通常不开放技术细节，可能调用第三方服务，不能定制机器人会话	精准、可控、复杂 意图识别+多轮对话+对接公开 API+知识图谱 领域意图和对话预先定义	用最短的对话轮次，满足用户需求
问答型	有任务目标 无需参数化请求	通常是一个平台，技术细节可能开放，可以让谱通用户配置修改机器人行业	较精确、较可控、较简单,需要自行挖掘问答对或补充同义问题 意图识别+多轮对话+对接企业 API+企业知识图谱	用最短的对话轮次，满足用户需求
闲聊型	开放,不限定领域	一家公司开发，直接面向C端消费者，通常不开放技术细节，可能调用第三方服务，很难定制机器人会话	几乎完全不可控,可直接调用 检索式:构建一个闲聊库，检索类似的问题，给出答案 生成式:从闲聊库这习生成模型	聊得越久越好


How to build a chatbot ?

你以为的搭建 Chatbot 的方式...

How to build a ChatBot Powered by Machine Learning ?

数学能力

VAE

$$
\begin{aligned}
&\ln p(x) \\
&= \ln \int_z p(x,z) \\
&= \ln \int_z p(x,z)\frac{q(z|x)}{q(z|x)} \\
&\geq \mathbb{E}_{q(z|x)}[\ln \frac{p(x,z)}{q(z|x)}] \\
&= \mathbb{E}_{q(z|x)}[\ln \frac{p(x|z)p(z)}{q(z|x)}] \\
&= \mathbb{E}_{q(z|x)}[\ln p(x|z)] + \mathbb{E}_{q(z|x)}[\ln \frac{p(z)}{q(z|x)}] \\
&= \mathbb{E}_{q(z|x)}[\ln p(x|z)] + \int_z q(z|x) \ln \frac{p(z)}{q(z|x)} \\
&= \mathbb{E}_{q(z|x)}[\ln p(x|z)] - D_{KL}[q(z|x)||p(z)] \\
&= likelihood - KL
\end{aligned}
$$

GAN

$$
\begin{aligned}
L &= E_{x\sim P_r}\left[\log D(x)\right] + E_{x\sim P_g}\left[\log\left(1 - D(x)\right)\right] \\
&= E_{x\sim P_r}\left[\log \frac{P_r(x)}{P_r(x)+P_g(x)}\right] + E_{x\sim P_g}\left[\log\left(1 - \frac{P_r(x)}{P_r(x)+P_g(x)}\right)\right] \\
&= E_{x\sim P_r}\left[\log\left(\frac{P_r(x)}{P_r(x)+P_g(x)}\right)\right] + E_{x\sim P_g}\left[\log\left(\frac{P_g(x)}{P_r(x)+P_g(x)}\right)\right] \\
&= E_{x\sim P_r}\left[\log\left(\frac{P_r(x)}{2 * \frac{1}{2}(P_r(x)+P_g(x))}\right)\right] + E_{x\sim P_g}\left[\log\left(\frac{P_g(x)}{2 * \frac{1}{2}(P_r(x)+P_g(x))}\right)\right] \\
&= E_{x\sim P_r}\left[\log\left(\frac{P_r(x)}{\frac{1}{2}(P_r(x)+P_g(x))}\right)\right] - \log 2 + E_{x\sim P_g}\left[\log\left(\frac{P_g(x)}{\frac{1}{2}(P_r(x)+P_g(x))}\right)\right] - log2 \\
&= E_{x\sim P_r}\left[\log\left(\frac{P_r(x)}{P_{average}(x)}\right)\right] + E_{x\sim P_g}\left[\log\left(\frac{P_g(x)}{P_{average}(x)}\right)\right] - 2log2 \\
&= 2JS(P_r|P_g) - 2\log 2
\end{aligned}
$$

概率论能力

$P(A|B) = \frac{P(A \cap B)}{P(B)}$

Law of Total Probability: If $A_1, \ldots, A_n$ *partitions* S, then $P(B) = P(B|A_1)P(A_1) + \cdots + P(B|A_n)P(A_n)$

Bayes Theorem: (con't from above) $P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}$

$E[X] = \sum_k k \cdot P(X = k)$ or $\int_{-\infty}^{\infty} x \cdot f(x)\, dx$, $\mathrm{Var}(X) = E[(X-\mu)^2] = E[X^2] - E[X]^2$

$E[aX + b] = aE[X] + b$, $E[X + Y] = E[X] + E[Y]$, $\mathrm{Var}(aX + b) = a^2\mathrm{Var}(X)$

$X \sim \mathrm{Ber}(p)$ $P(X = 0) = 1 - p,\ P(X = 1) = p$, $E[X] = p$, $\mathrm{Var}(X) = p(1-p)$

$X \sim \mathrm{Bin}(n, p)$ $P(X = k) = \binom{n}{k} p^k (1-p)^{n-k},\ k = 0, 1, \ldots, n$, $E[X] = np$, $\mathrm{Var}(X) = np(1-p)$

$X \sim \mathrm{Geo}(p)$ $P(X = k) = (1-p)^{k-1}p,\ k = 1, 2, \ldots$, $E[X] = 1/p$, $\mathrm{Var}(X) = (1-p)/p^2$

$X \sim \mathrm{Poi}(\lambda)$ $P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!},\ k = 0, 1, 2, \ldots$, $E[X] = \lambda$, $\mathrm{Var}(X) = \lambda$

$X \sim \mathrm{U}(a, b)$ $f(x) = \frac{1}{b-a},\ a < x < b$, $E[X] = \frac{b+a}{2}$, $\mathrm{Var}(X) = \frac{(b-a)^2}{12}$

$X \sim \mathrm{Expo}(\lambda)$ $f(x) = \lambda e^{-\lambda x},\ x > 0$, $E[X] = \frac{1}{\lambda}$, $\mathrm{Var}(X) = \frac{1}{\lambda^2}$

$X \sim \mathrm{N}(\mu, \sigma^2)$ $f(x) = \text{not needed}$, $E[X] = \mu$, $\mathrm{Var}(X) = \sigma^2$

If $X \sim N(\mu, \sigma^2)$, then $\frac{X-\mu}{\sigma} \sim N(0, 1)$

Sample statistics: $\bar{X}_n = \frac{1}{n}\sum X_i$, $S^2 = \frac{1}{n-1}\sum (X_i - \bar{X}_n)^2$

For X_i's i.i.d. $\sim \mathrm{N}(\mu, \sigma^2)$, σ known, $\bar{X}_n \pm z_{\alpha/2}\frac{\sigma}{\sqrt{n}}$ is a $100(1-\alpha)\%$ CI for μ

ANOVA: $S_i^2 = \frac{1}{J-1}\sum (X_{ij} - \bar{X}_{i\cdot})^2$, $\mathrm{MSTr} = \frac{J}{I-1}\sum (\bar{X}_{i\cdot} - \bar{X}_{\cdot\cdot})^2$, $\mathrm{MSE} = \frac{1}{I}\sum S_i^2$, $F = \frac{\mathrm{MSTr}}{\mathrm{MSE}}$

算法能力

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

软件能力

```
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
# evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = model_selection.KFold(n_splits=10, random_state=seed)
    cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

工程能力

Tokenization

Training examples	love	programming	
1 → I love programming	1	1	
2 → Programming also loves me	1		1

Documents

NLTK

Lemmatization

Stopwords

Natural language processing

Source: Natural Language Processing

阅读论文

arXiv:1706.03762v5 [cs.CL] 6 Dec 2017

Attention Is All You Need

Ashish Vaswani[*] Google Brain avaswani@google.com
Noam Shazeer[*] Google Brain noam@google.com
Niki Parmar[*] Google Research nikip@google.com
Jakob Uszkoreit[*] Google Research usz@google.com
Llion Jones[*] Google Research llion@google.com
Aidan N. Gomez[*] [†] University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser[*] Google Brain lukaszkaiser@google.com
Illia Polosukhin[*] [‡] illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

In this work we propose the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as twelve hours on eight P100 GPUs.

2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

3 Model Architecture

Most competitive neural sequence transduction models have an encoder-decoder structure [5, 2, 35]. Here, the encoder maps an input sequence of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations $\mathbf{z} = (z_1, ..., z_n)$. Given $\mathbf{z}$, the decoder then generates an output sequence $(y_1, ..., y_m)$ of symbols one element at a time. At each step the model is auto-regressive [10], consuming the previously generated symbols as additional input when generating the next.

The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 1, respectively.

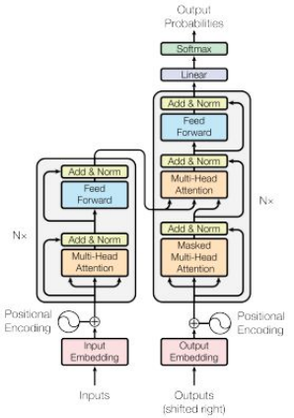


Figure 1: The Transformer - model architecture.

3.1 Encoder and Decoder Stacks

Encoder: The encoder is composed of a stack of $N = 6$ identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. We employ a residual connection [11] around each of the two sub-layers, followed by layer normalization [1]. That is, the output of each sub-layer is $\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$, where $\mathrm{Sublayer}(x)$ is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension $d_{\text{model}} = 512$.

Decoder: The decoder is also composed of a stack of $N = 6$ identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we employ residual connections around each of the sub-layers, followed by layer normalization. We also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i.

3.2 Attention

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

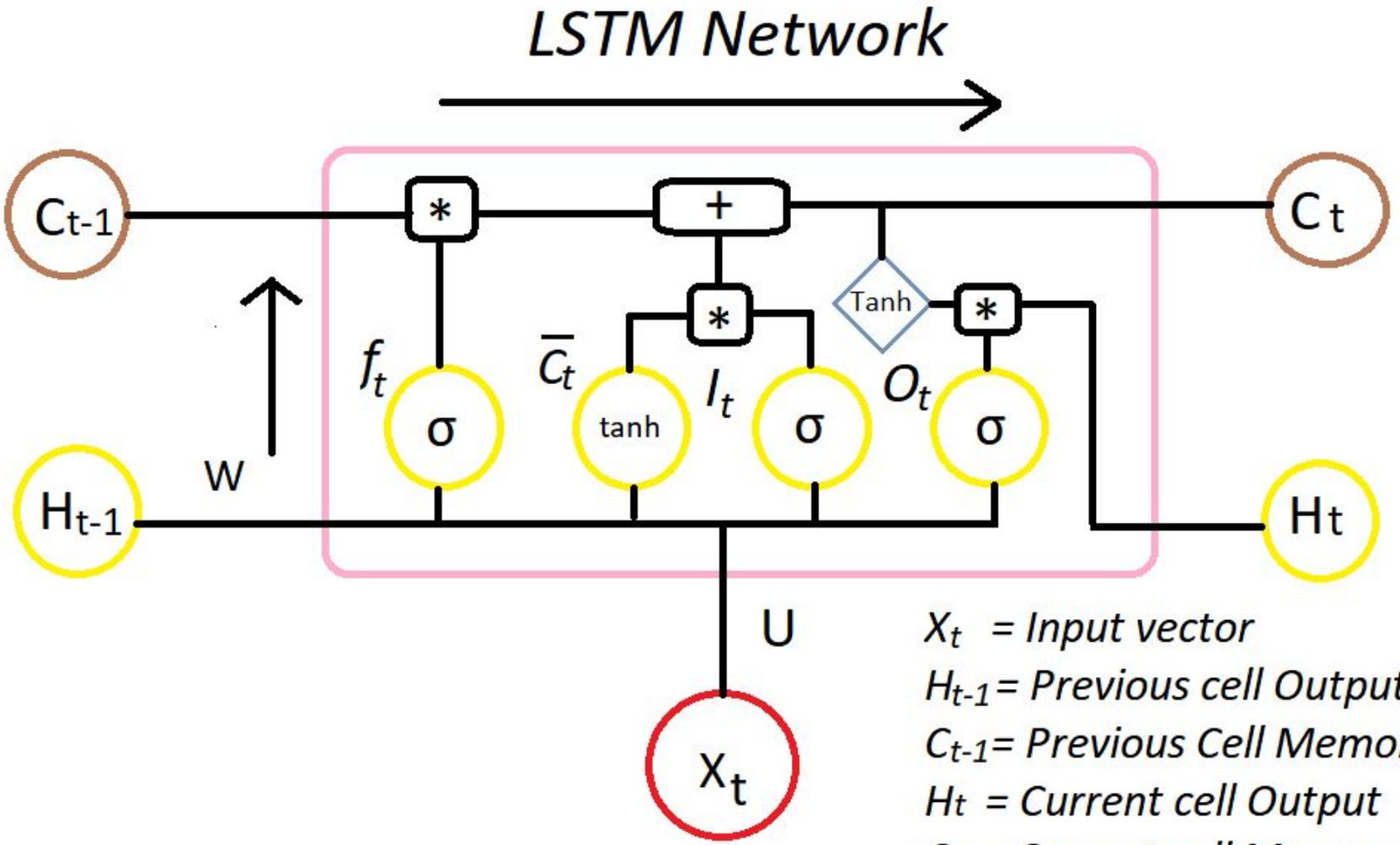
[*] Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†] Work performed while at Google Brain.

[‡] Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

了解神经元



X_t = Input vector
H_{t-1} = Previous cell Output
C_{t-1} = Previous Cell Memory
H_t = Current cell Output
C_t = Current cell Memory

$*$ = Element-wise multiplication
$+$ = Element-wise addition

$$f_t = \sigma (X_t * U_f + H_{t-1} * W_f)$$
$$\bar{C}_t = tanh (X_t * U_c + H_{t-1} * W_c)$$
$$I_t = \sigma (X_t * U_i + H_{t-1} * W_i)$$
$$O_t = \sigma (X_t * U_o + H_{t-1} * W_o)$$

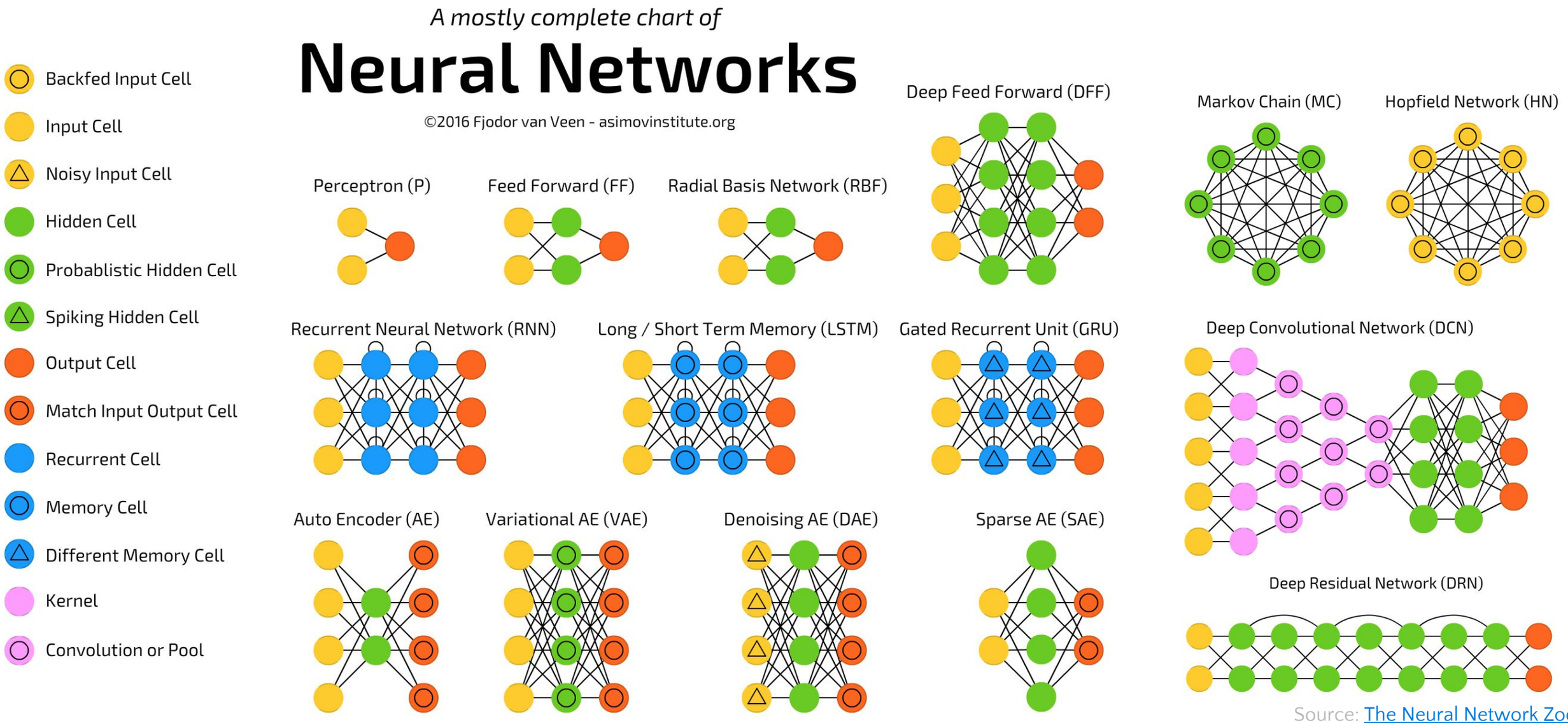
$$C_t = f_t * C_{t-1} + I_t * \bar{C}_t$$
$$H_t = O_t * tanh (C_t)$$

W, U = weight vectors for forget gate (f), candidate (c) , i/p gate (I) and o/p gate (O)

Note : These are different weights for different gates, for simpicity's sake, I mentioned W and U

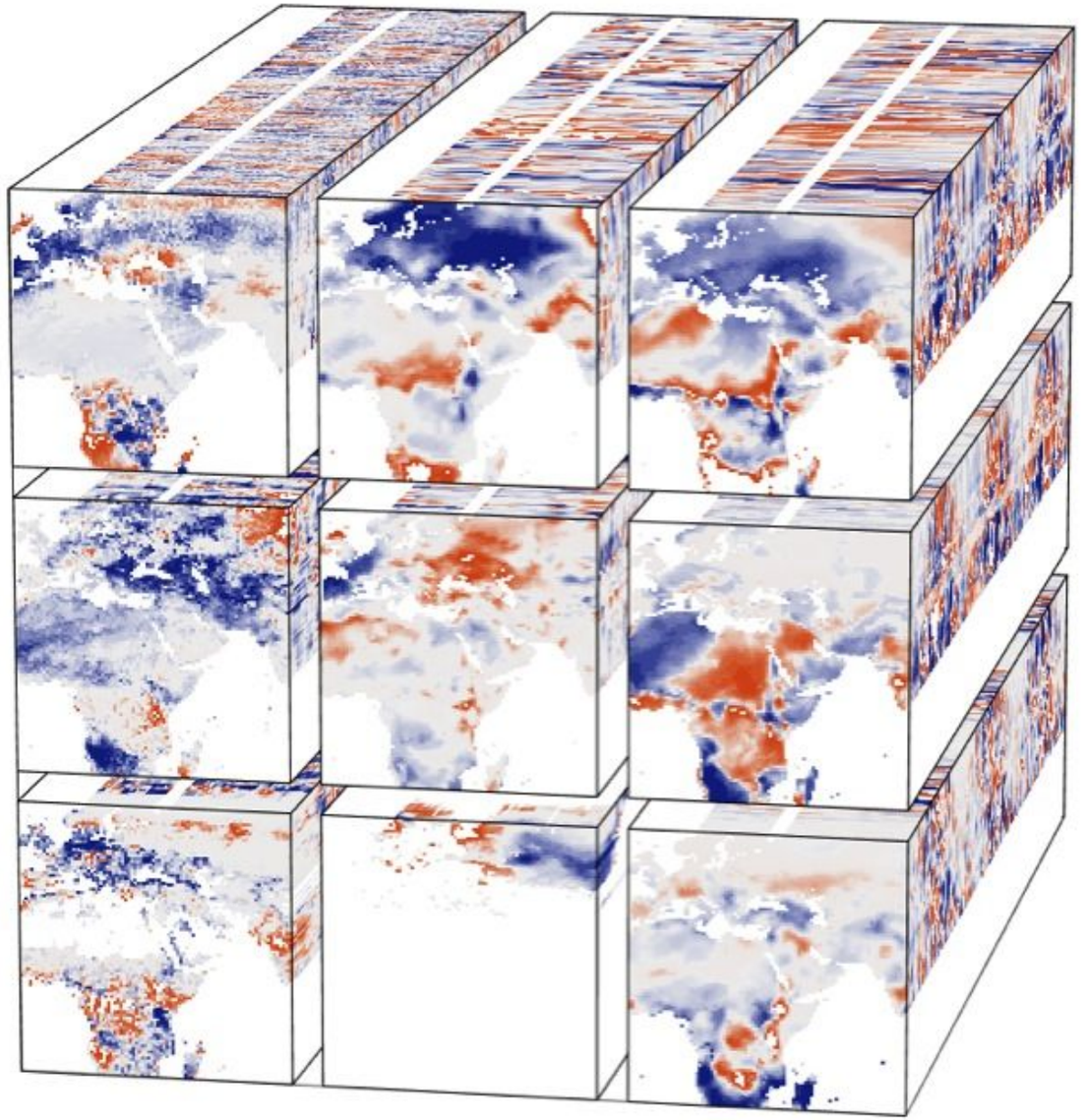
Source: DeepNLP – LSTM with Math.

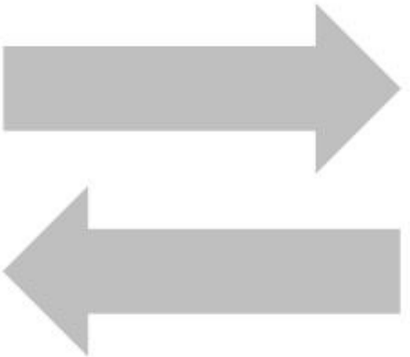
分析各种神经网络结构

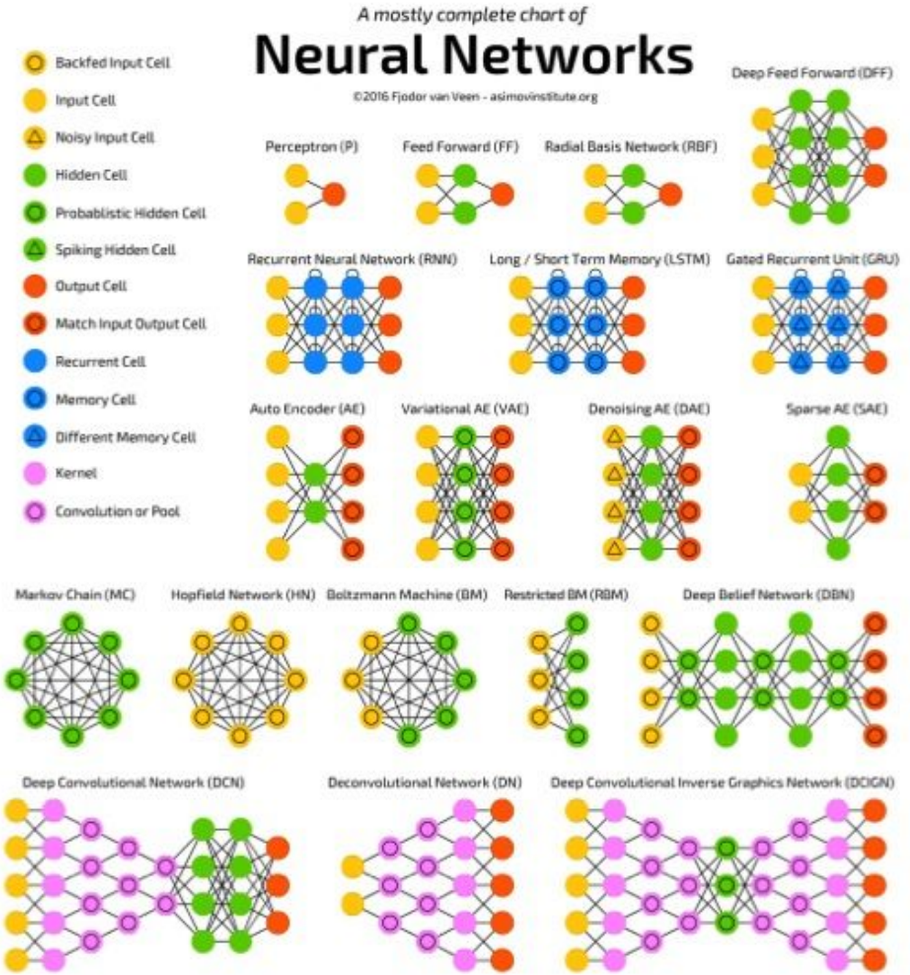


Source: [The Neural Network Zoo]

高维空间抽象思维能力







Source: [Earth System Data Cube]

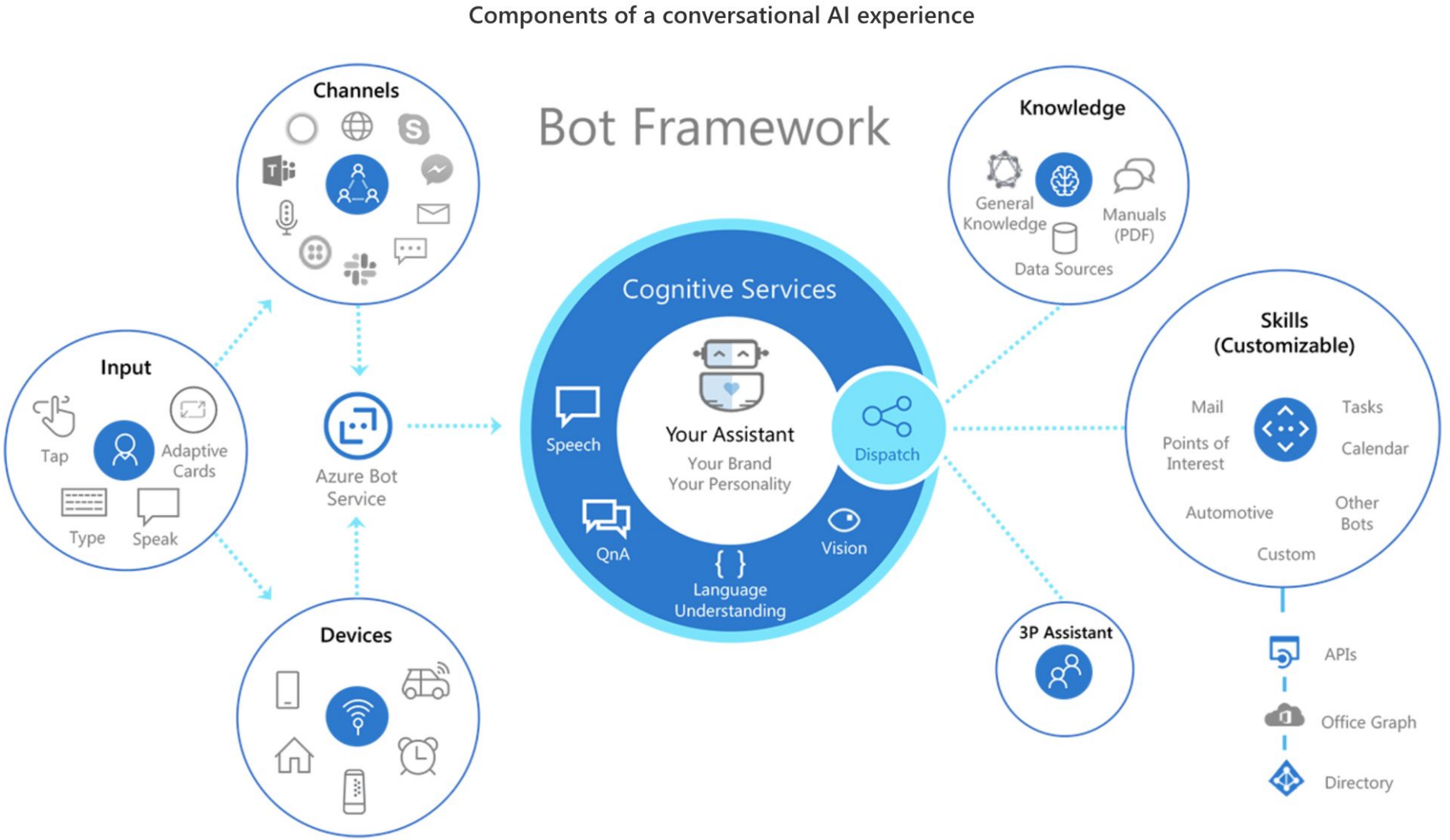
Source: WTF?! How future humans will ...

聊天机器人从入门到放弃

全剧终

搭建 Chatbot 的正确姿势

使用 Microsoft Bot Framework



推荐阅读

由课题技术导师 **李佳芮、李卓桓** 编著
国内对话式交互设计畅销书

- 人工智能与 Chatbot 应用场景
- 构建 Chatbot 应用的案例剖析和最佳实战
- 展望基于人工智能开发 Chatbot 的机会

“未来，Chatbot将成为重要的人机交互窗口。这一切，都可以从设计一个小的封闭域Chatbot开始，从《Chatbot从0到1：对话式交互设计实践指南》开始。”

陆 奇

奇绩创坛（前YC中国）创始人，句子互动导师

前百度总裁，前微软任执行副总裁

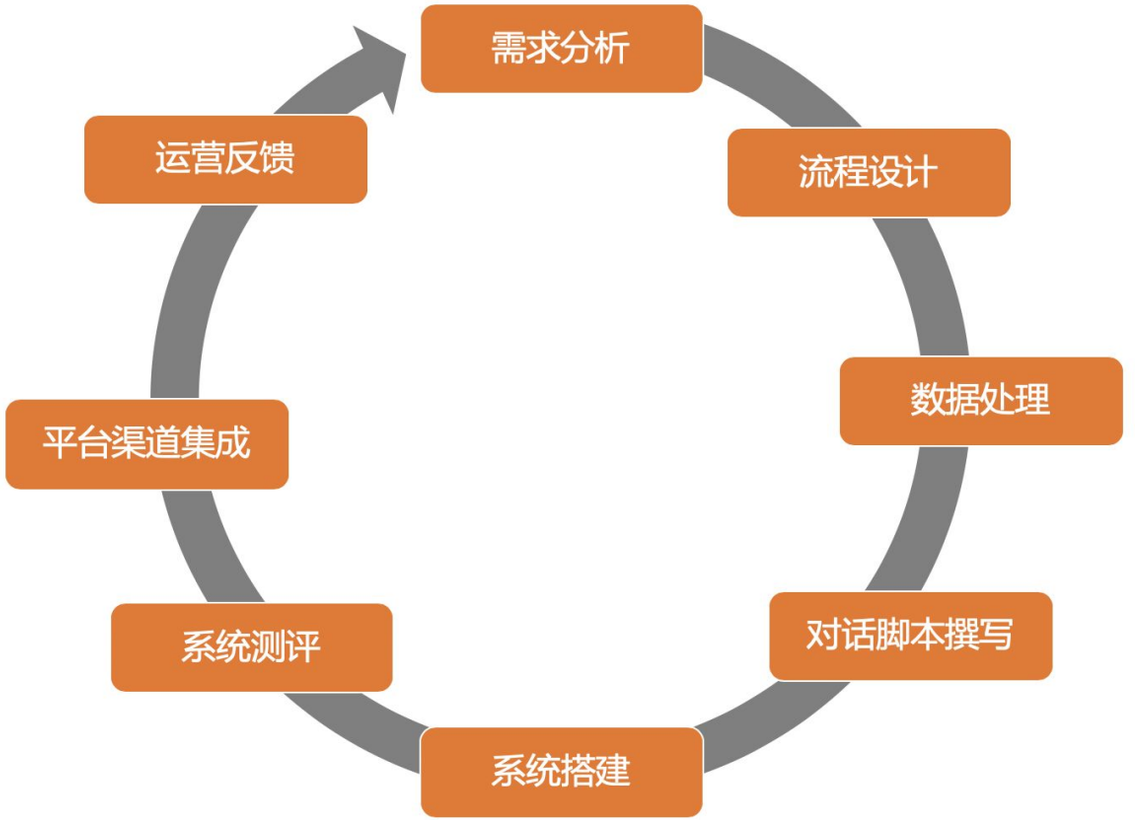
奇绩创坛 MIRACLEPLUS

句子互动 Juzi.bot

《Chatbot 从0到1》

京东地址：https://item.jd.com/12630213.html

第1部分从人工智能的发展带动对话式交互引出Chatbot的应用场景及其分类；第2部分和第3部分从需求分析、流程设计、数据处理、对话脚本撰写、系统搭建、对话任务测评、平台渠道集成、运营反馈等方面，对Chatbot的整个生命周期进行了详细分析和介绍；第4部分通过案例分析，对Chatbot进行了实践；第5部分总结了目前对话式交互的局限性，并展望了基于人工智能发展Chatbot的机会。



《Chatbot从0到1：对话式交互设计实践指南》适合希望从事Chatbot行业的读者阅读，尤其是正在考虑将业务切入 Chatbot领域的决策者，即将或正在从事Chatbot专业工作的产品经理和项目经理，以及希望了解Chatbot领域工作流程的开发人员。

1. 需求分析

1 确定 Chatbot 的边界
2 确定 Chatbot 形象
3 “六何”分析法
4 案例: 差旅 Chatbot

2. 流程设计

1 对话流程设计的原则

    1.1 不能保证 Chatbot 成功的因素

    1.2 影响 Chatbot 成功的因素

2 梳理业务要素

3 抽取对话流程, 绘制流程图

    3.1 绘制基础业务流程图

    3.2 绘制跨职能流程图

    3.3 进行业务线合并

3. 数据处理

1 数据收集
2 数据扩充
3 数据清理
4 数据标注

4. 对话脚本撰写

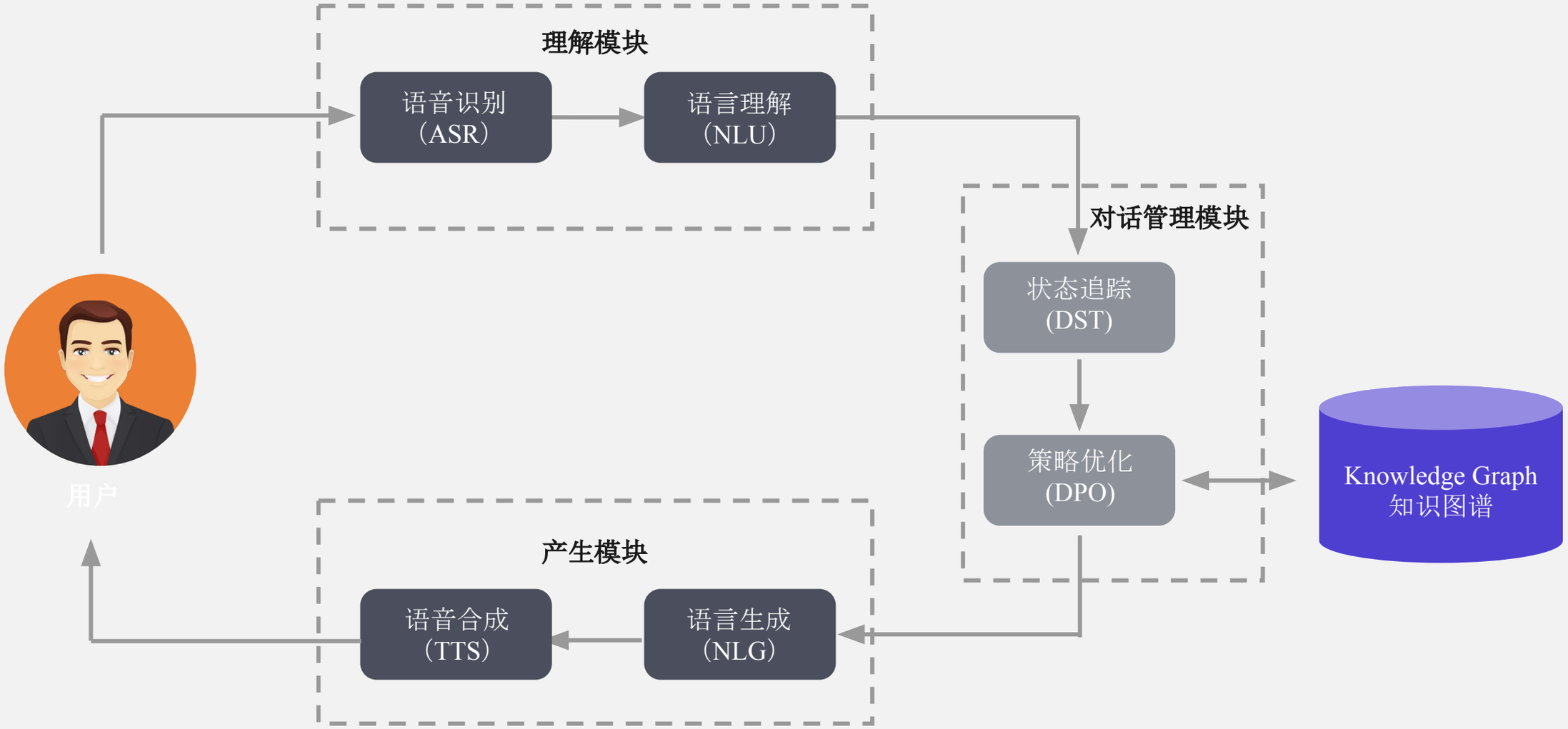
十条设计原则：

1 简洁明了
2 对话语句要自然
3 区别新老用户
4 使用问候语和结束语
5 确认策略
6 随机策略
7 使用对话式标识
8 设计延迟话术
9 主动学习
10 持续跟踪上下文

控制对话流：

1 首次互动
2 持续引导
3 处理中断
4 设计导航

5. 对话系统搭建


理解模块
语音识别
（ASR）
语言理解
（NLU）
对话管理模块
状态追踪
(DST)
策略优化
(DPO)
Knowledge Graph
知识图谱
用户
产生模块
语音合成
（TTS）
语言生成
（NLG）

6. 对话任务测评

任务型 Chatbot 的测评

1 自然语言理解测评

2 对话管理测评

3 自然语言生成测评

问答型 Chatbot 的测评

1 准确率

2 召回率

3 F值

4 问题解决率

闲聊型 Chatbot 的测评

1 线上指标

2 客观评价

3 主观评价

7. 平台渠道集成

8. 运营反馈

流量分析

1. 对话流量
2. 会话流量
3. 平台渠道分析

对话内容分析

1 对话聚类

2 意图统计

3 消息漏斗

4 词云分析

5 情绪分析

6 用户给Chatbot 的评分

7 转化路径分析

对话异常分析

1 异常对话记录

2 热门退出消息

3 调用默认回答的次数

用户分析

1 活跃用户

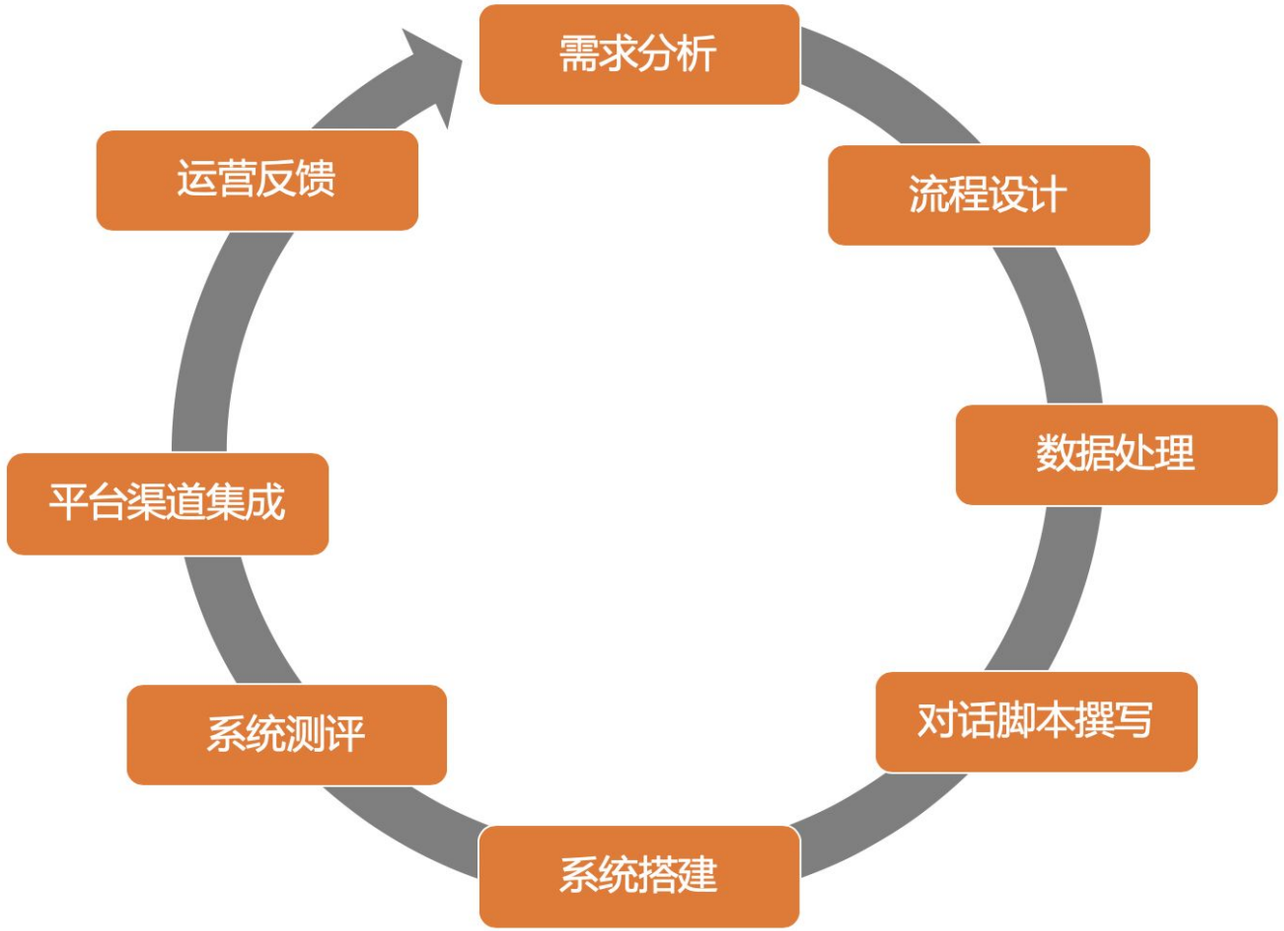
2 用户留存率

3 用户活跃度

4 用户总数

5 用户画像

Chatbot 的八大生命周期



商业应用

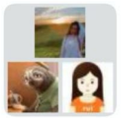
搭建一个CEO助理机器人

商业应用举例

1. 保险助理
2. 智能营销
3. 内部协同
4. 售后服务
5. ….

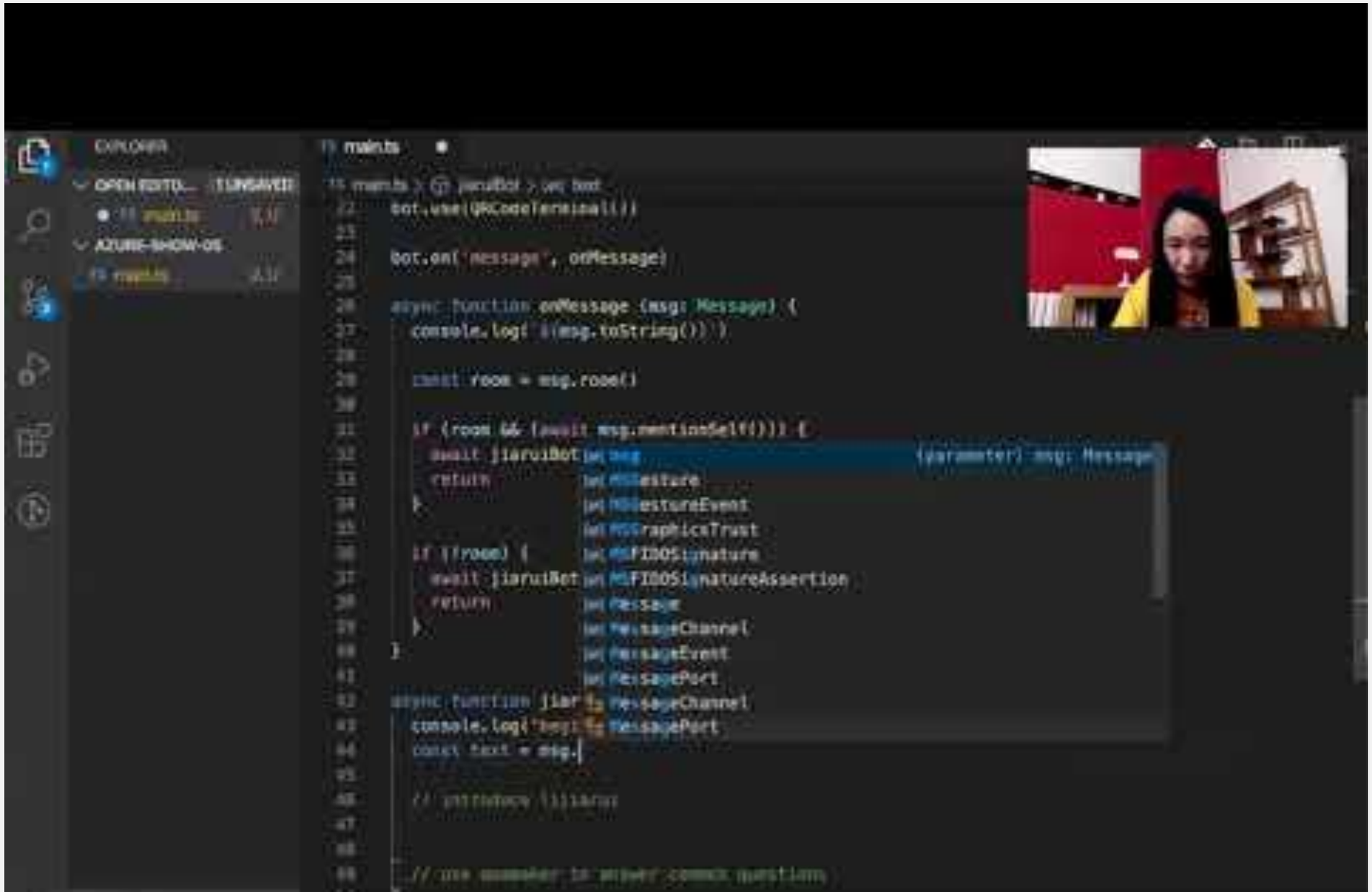
DEMO

立即扫码入群@机器人体验



清华+微软人工智能商业应用

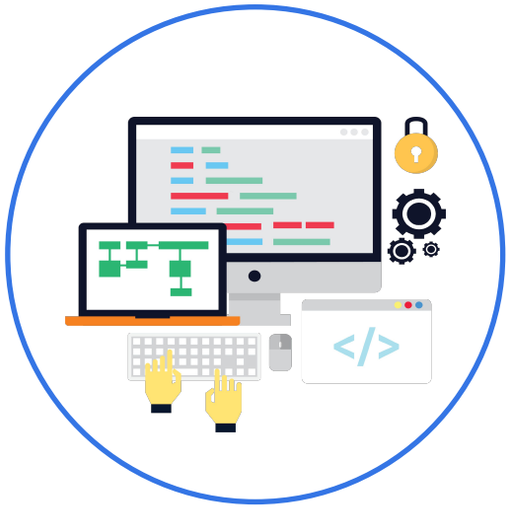
该二维码7天内(6月8日前)有效，重新进入将更新



1. 如何快速接入微信个人号
2. 使用 QnAMaker

Connecting Chatbots

A **Conversational RPA SDK** for **Wechat** which can help you create a bot in 6 lines of **javascript**, with cross-platform support including **Linux**, **Windows**, **Darwin**(**OSX**/**Mac**) and **Docker**.



Developer

wechaty

Chatbot Connector

WeChat ChatBot

npm package 0.28.3 | downloads 4.9k/month | github stars 6.3k | docker pulls 31k | </> TypeScript | Greenkeeper enabled | chat on gitter

Wechaty 是一个帮助你自动化消息处理流程的 RPA 工具

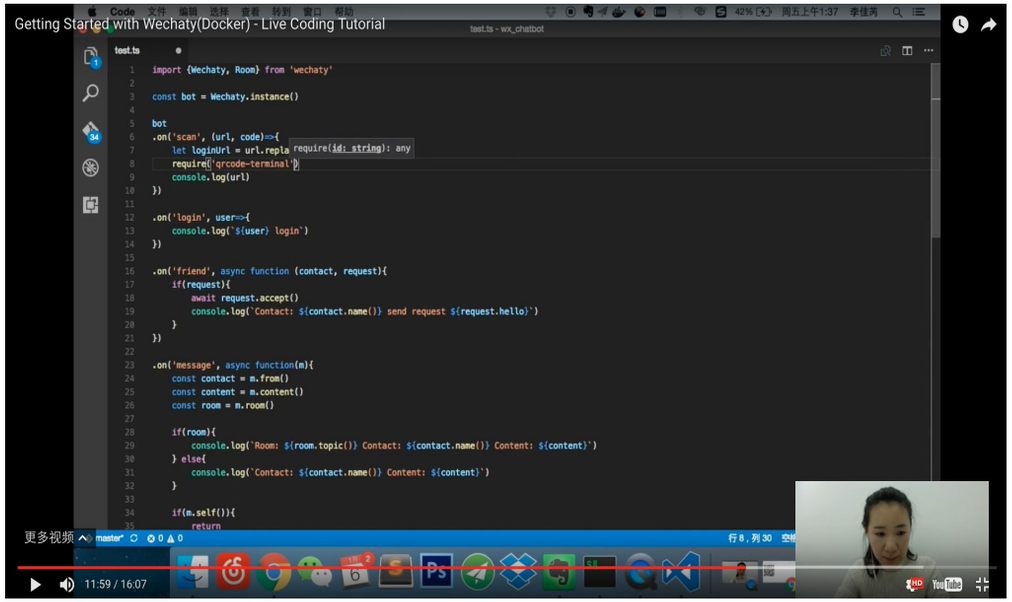
它支持的功能包括：接收消息，发送消息、添加好友、为好友备注、接受好友请求、发起群聊、加人入群等功能。

只需要6行代码，你就可以 通过个人号 搭建一个 微信机器人功能，用来自动管理微信消息。

更多功能包括：

- 消息处理：关键词回复
- 群管理：自动入群，拉人，踢人
- 自动处理好友请求
- 智能对话：通过简单配置，即可加入智能对话系统，完成指定任务

The World's Shortest Chatbot Code




```
const { Wechaty } = require('wechaty') // import Wechaty from 'wechaty'

Wechaty.instance() // Singleton
.on('scan', (url, code) => console.log(`Scan QR Code to login: ${code}\n${url}`))
.on('login',       user => console.log(`User ${user} logined`))
.on('message',  message => console.log(`Message: ${message}`))
.init()
```

https://github.com/wechaty/wechaty
https://github.com/wechaty/wechaty-getting-started

Used by hundreds of Projects

Chatie / wechaty

Sponsor | Used by 415 | Unwatch 266 | Unstar 6,298 | Fork 935

Code | Issues 122 | 10 | Wiki | Security | Insights

415

- Pulse
- Contributors
- Community
- Traffic
- Commits
- Code frequency
- **Dependency graph**
- Network
- Forks

Dependencies | Dependents

Repositories that depend on **wechaty**

415 Repositories | 15 Packages

gengchen528 / **wechatBot**	★ 1,007 ⑂ 197
t9tio / **wewe**	★ 205 ⑂ 16
gengchen528 / **wechat-assistant**	★ 156 ⑂ 43
UnsignedInt8 / **leavexchat-bot**	★ 29 ⑂ 3
dotnetclub-net / **club-chaty**	★ 28 ⑂ 3
liuxing / **node-abc**	★ 273 ⑂ 86
ningowood / **aweningo-chat-bot**	★ 14 ⑂ 1
Bobolovewill / **WechatHelper**	★ 21 ⑂ 0
coderwhocode / **wechaty-pay**	★ 18 ⑂ 1
Mcbai / **WeChat-bot**	★ 27 ⑂ 11
BingKui / **WeChatRobot**	★ 26 ⑂ 3
felixrieseberg / **slack-wechat-bridge**	★ 1 ⑂ 0

LeaveXChat

使用 Telegram Bot 接收 WeChat 文字、语音、图片、视频消息

Slack - WeChat Bridge

This is Node app that allows you to post messages in Slack Channels to a WeChat g
WeChat group to a Slack channel. It's essentially a bridge. It works well, but has har

- A WeChat account created before July 2017 (WeChat has turned off access to
  after)
- A hosting environment that supports puppeteer, Google's solution for controllin

Aweningo-Chat-Bot 群聊机器人助手

> 前景回顾：本项目前身为 wechat-go，一个简单的微信群聊机器人启动项目 改版说明：由于微信机器人生态
> 以及部分个人原因，此项目需要结合未来发展，重新构建为一个专注收集各类常用群聊工具机器人教程的资
> 库，并逐步融合到 aweningo 体系中。

dotnet club chaty

在微信里聊的嗨翻天之后，微信里的聊天内容其实有更多价值。你可能希望珍藏这些记忆，可能希望转发给更多互联网上的人
分享……

dotnet club 有一样的需求，我们希望把人们散落在各个技术群里的讨论沉淀下来，留给没有参与
讨论的效用更深远，所以我们开发了这个项目。

dotnet club chaty 是帮助你导出这些聊天记录的利器：

- 用微信个人号登录
- 就像与好友聊天那样简单
- 导出的结果可以用于分享、打印
- 结构化的数据，方便二次使用

WechatHelper

很听你话的私人宝宝波的专属助手，帮你创建定时任务，每日提醒，

文字支持格式：（关键词之间需要用空格分开，）

- “提醒 我 18:30 快要下班了，准备一下，不要忘记带东西”（当
- “提醒 其他人昵称 2019-09-10 10:00 工作再忙，也要记得喝水
- “提醒 我 每天 8:00 出门记得带钥匙，公交卡还有饭盒”（每日
- “提醒 wo 2019-09-10 10:00 还有两天就是女票的生日，要提

wewe

Open group chat messages to the world

Join the community of t9t.io | Get in touch | Powered By Wechaty

Core values of wewe

- Open group chat to the internet
- Search engine friendly
- Extract topics from message history

A Wechaty bot to make girlfriend happy: 1,000+ star

bilibili

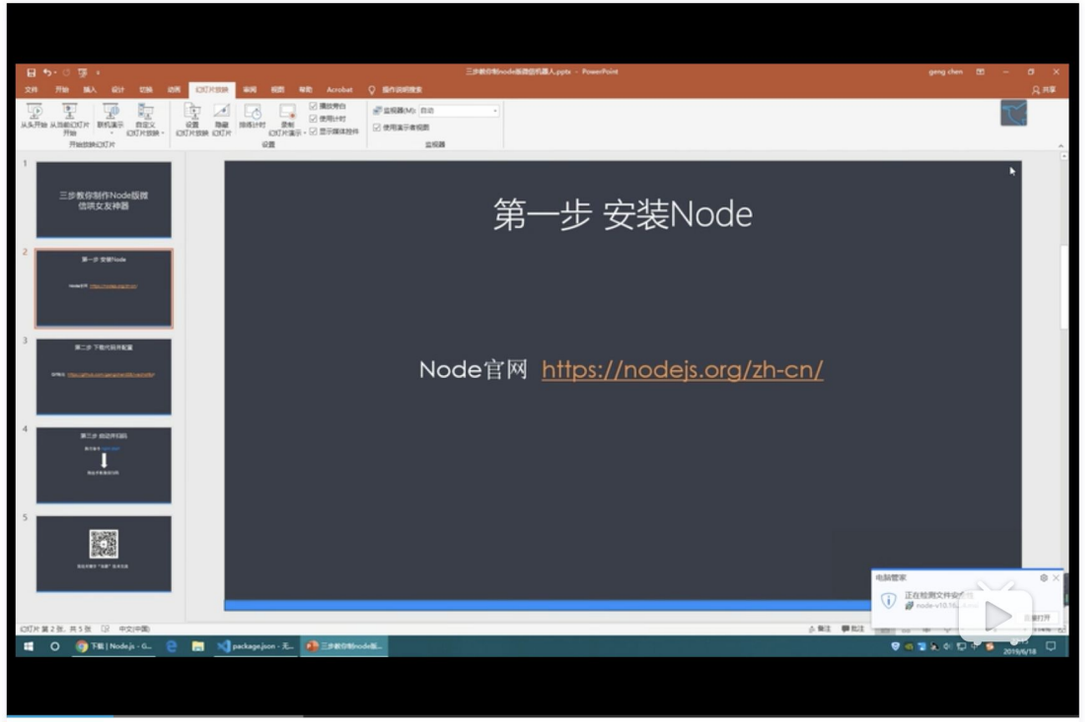
主站 音频 游戏中心 直播 会员购 漫画 BW 70年 下载APP

用程序员的格子衬衫做了lo裙

三步教你用Node做一个微信哄女友神器

生活 > 日常 2019-06-19 14:25:52

2580播放 · 2弹幕 未经作者授权，禁止转载



Source: https://www.bilibili.com/video/av56077628

gengchen528 / wechatBot

Unwatch 29 | Unstar 1,007 | Fork 200

Code | Issues 1 | Pull requests 0 | Projects 0 | Wiki | Security | Insights

微信每日说，三步教你用Node做一个微信哄女友(基友)神器！还能帮女朋友解决垃圾分类难题

51 commits | 3 branches | 0 releases | 1 contributor | MIT

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

README.md

微信每日说

node >=10 | wechaty >=0.27 | Window | Mac | Centos

wechatBot是基于node与wechaty的微信小情话工具。最初功能只有每日发送天气和一句情话，后来添加了智能机器人聊天功能，自动加群，自动加好友，定时助手功能等。但由于本项目面向小白用户与刚接触node开发的用户，故拆分了两个项目，一个是功能专一面向小白的《微信每日说》（也就是本项目），另一个也在我的仓库下《微信个人秘书》面向有较多编程经验的用户。下面主要介绍微信每日说的使用

主要功能

- 定时给女朋友发送每日天气提醒，以及每日一句
- 天行机器人自动陪女朋友聊天（需要自己申请天行机器人api，不过目前开源的机器人api都不要抱太大希望，因为很傻的，如果你有发现好的机器人可以来推荐）
- 垃圾分类功能，使用方法：？ 垃圾名称
- 最近看到python版支持多女朋友配置，我思考了一下，还是不要加了比较好，我们要做一个专一的人，哈哈
- 想要更多功能，请移步《微信个人秘书》

Source: https://github.com/gengchen528/wechatBot

Projects Using Wechaty

1. koa+wechaty实现的微信个人秘书，把你闲置的微信号利用起来做个个人秘书
2. 微信每日说，每日自动发送微信消息
3. 一个通过深度神经网络CNN给图片评分的wechaty项目
4. Use Wechaty to get notifications when you receive new emails
5. RoomAnalyze
6. Wechaty 群分析统计机器人
7. Poker planning for scrum using wechat bot
8. help to invite people to a group
9. Philly Wechat Wechat <=> philly-wechat <=> philly
10. Haoshiyou-Bot A chat bot supported on WeChaty, managing the HaoShiYou wechat groups run by volunteers of haoshiyou.org
11. A microservice implementation of WeChaty
12. Wechaty Telegram Bot Adaptor Run your Telegram bot on WeChat
13. Bot Builder Wechaty Connector Connector for UniversalBot from Bot Builder
14. 命令行版的微信

Source: https://github.com/Chatie/wechaty/wiki/Projects-Using-Wechaty

Hundreds of blog from developers

Wechaty - Conversational RPA SDK for WeChat

Posts Categories

Wechaty

Bot is not born, but made.

Website

Twitter

GitHub

Recent Posts

repe-assistant 社群活动助手

less than 1 minute read

Author: @xiaogan18 Full stack developer,specialized golang and blockchain. Code: @Github

Cps demo of wechat bot with wechaty

4 minute read

Nio bot蔚来车主群服务机器人

1 minute read

作者: Leons828 web developer Code: Github

Chatbot 在定制旅游行业的应用

less than 1 minute read

作者: 李恺，无二之旅研发总监、资深后端工程师。

Posts by Category

project	19	event	13	tutorial	11
announcement	6	article	6	story	5
feature	4	hack	4	npm	2
migration	2	shop	1		

Posts by Tag

code	34	meetup	9	talk	8
release	3	interview	2	docker	2
startup	2	api	2	machine learning	2
hook	2	datagirls	2	artificial intelligence	2
bot	2	case	1	sticker	1
telegram	1	unofficial	1	document	1
typescript	1	botbuilder	1	sdk	1
windows	1	install	1	heroku	1
deploy	1	analytics	1	contribution	1
open source	1	devops	1	npm	1
saas	1	bot5	1	rasa	1
bot friday	1	dotnetclub	1		

1. 如何快速接入微信个人号
2. **使用 QnAMaker**

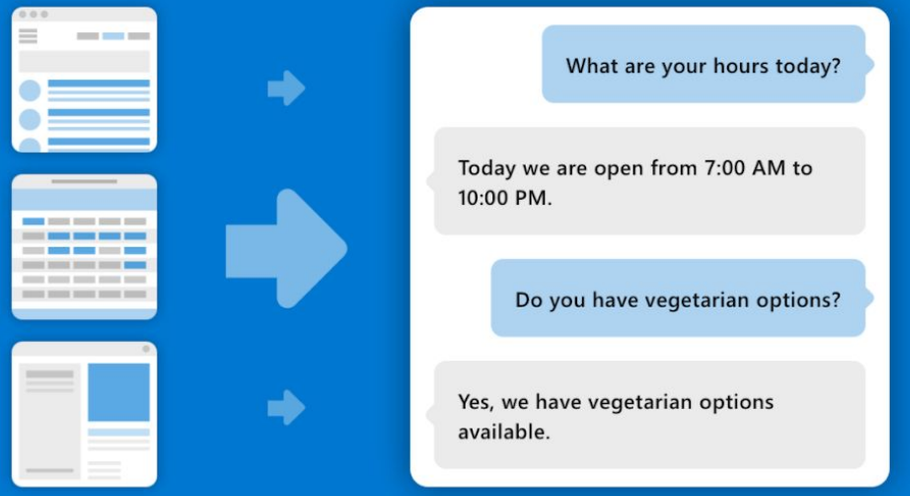
Microsoft Azure | Cognitive Services | QnA Maker Support Resources Sign in

Design sophisticated multi-turn conversations easily with follow-up prompts. Learn more.

From data to bot in minutes

Build, train and publish a sophisticated bot using FAQ pages, support websites, product manuals, SharePoint documents or editorial content through an easy-to-use UI or via REST APIs.

Get started >

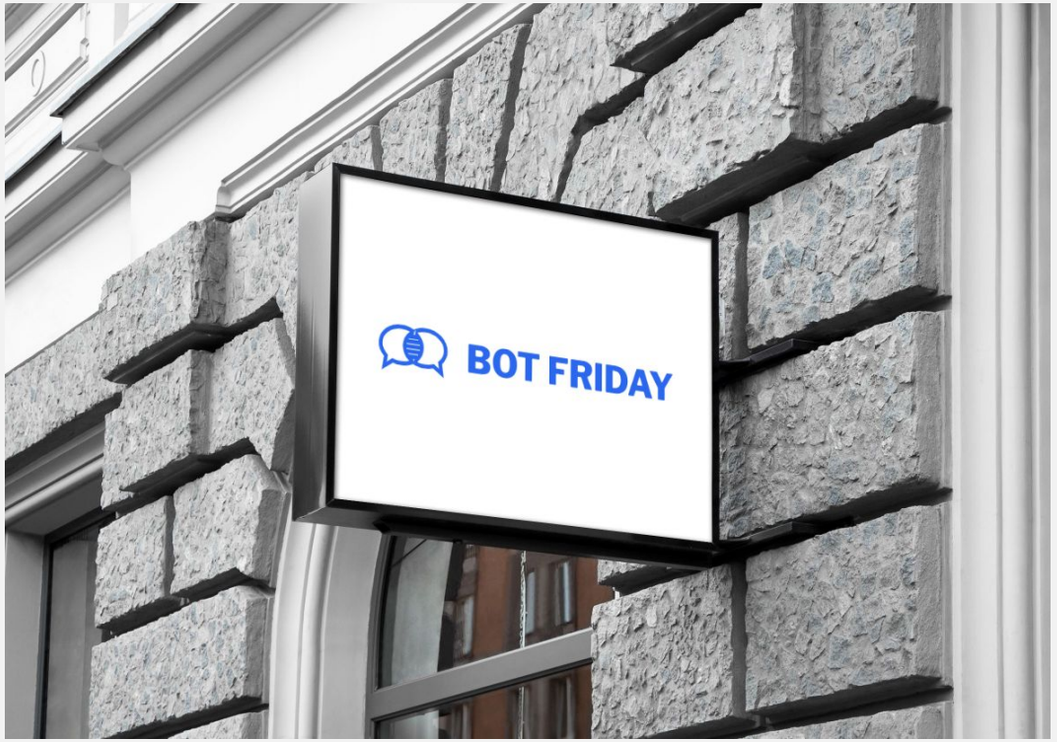


动手实验


Code
Time !
Welcome
{} package.json
TS mybot.ts
JS mybot.js
if (/hello/.test(content)) {
m.say("how are you")
}
if (/room/.test(content)) {
let keyroom = await Room.find({ topic: "test" })
if (keyroom) {
await keyroom.add(contact)
await keyroom.say("welcome!", contact)
}
}

更多线下讨论

Bot Friday Club



- 19:00 - 22:00 Chatbot Talk every Friaday
- After Party
- 50 talk each year
- One Chatbot bootcamp each year all over the world
- https://www.bot5.club/manuals/newcomer/

Bot Friday Activities

《The Bad Part of My Chatbot Experience》
《Face recognition chatbot based on WeChat》

Chatbot Developer	3
Chatbot Product Manager	1
Chatbot Founder	8
Total	12

Bot Friday seminar 2

《Chatbot Design Principle》
《Chatbot getting started》
《Emotional ChatBot》

Chatbot Developer	7
Chatbot Product Manager	3
Chatbot Founder	8
AI investor	1
Total	19

Bot Friday seminar 3

《Bot Framework introduction》
《How to use Microsoft cognitive service build a bot》
《WeChat Adapter for bot framework》

Tencent Chatbot Product Manager	3
Microsoft Chatbot Product Manager	5
Chatbot Founder	4
Chatbot Developer	4
Total	16

Bot Friday seminar 6

Bot Friday Club 报名方法

新人参加活动要求

1. 通过一位 BOT5 俱乐部的正式会员推荐。在未来推荐人将作为新人的 Mentor，Mentor需要帮助新人加入 Bot Friday Open Forum 微信群中。
2. **watch** & **star** 沙龙活动组织的 Repository: https://github.com/wechaty/bot5.club
3. 新人第一次加入之前，需要学习 Bot Friday Club 的活动规则、目标、规则和背景：
    1. （必读）阅读最后一场沙龙活动纪要
    2. （可选）阅读沙龙活动手册
4. 新人第一次参加方法：在最后一场沙龙活动纪要中的评论区按照下面模板进行报名：
    1. 姓名
    2. 公司
    3. 个人简介（100字左右即可）
    4. 推荐人
    5. 分享内容（如果有）
5. 如果沙龙活动人数过多，主席有权限将总人数限制在主席认为的合理范围之内。新人有可能会被拒绝参加活动。

脸盲助手，您身边的智能认人助手!
长按识别二维码，告别脸盲。

Microsoft

清華大學 Tsinghua University

从0到1搭建对话机器人

项目库地址： https://github.com/lijiarui/chatbot-zero-to-one/

具体代码在

：https://github.com/lijiarui/chatbot-zero-to-one/tree/master/live-coding/azure-show-05

李佳芮

Microsoft AI MVP

Founder & CEO of JuziBot

MVP

Microsoft®

Most Valuable Professional

Microsoft

清華大學 Tsinghua University

Thanks & Q/A

李佳芮

Microsoft AI MVP

Founder & CEO of JuziBot

MVP

Microsoft®

Most Valuable Professional