

建立属于你的智能客服

背景

很多人问，对话式交互系统就是语音交互么？当然不是。语音交互本身真的算不上新概念，大家可能都给银行打过电话，“普通话服务请按1，英文服务请按2.....返回上一层请按0”这也算对话式交互系统，我想大家都清楚这种交互带来的用户体验有多低效。那么对话式交互系统已经可以取代人类提供服务了么？也没有，图灵测试还没有过呢，着什么急啊。

不过，随着人工智能的发展，对话式交互穿着语音和文本的外衣，携手模糊搜索引擎，怀抱计算科学和语言学的内核，带着定制化推荐的花环，驾着深度学习和大数据的马车乘风破浪而来——我们就知道，大约是时候了。至少，我们已经可以十分钟内创造自己的对话式客服了。今天的文章大约分三章，历史，今天（chatbot api）和未来（深度学习和智能问答）。

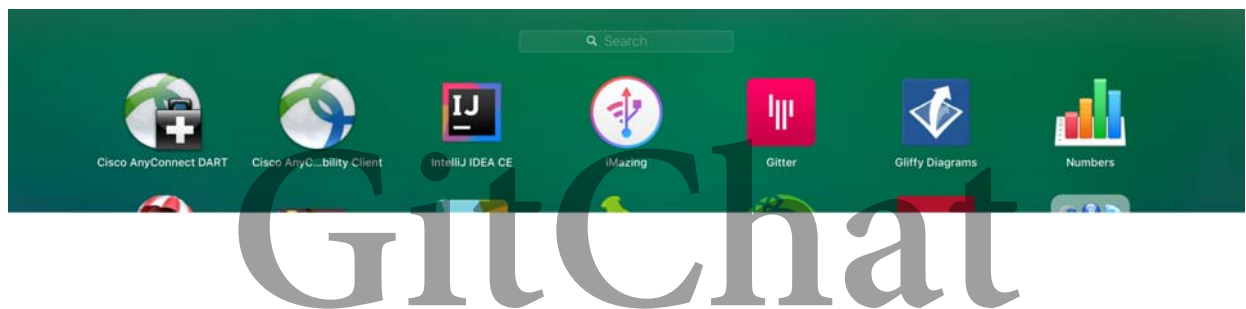
先定义一下交互系统，wiki给出的定义是“Interaction is a kind of action that occurs as two or more objects have an effect upon one another.”，也就是说双方或者多方相互影响的过程，那么在咱们的上下文里面，我们不妨限定为人机交互。先来讲讲是什么，再来讲讲怎么做吧。

历史和现在

广义上的对话式交互实际上包括所有一问一答形式的人机交互，自始至终，我们都需要从机器拿到信息。在最早的时代用的是文本交互系统TUI，其实直到今天我相信程序员们在Linux下面完成大部分操作还是会选用Terminal，这种文本交互非常简洁高效，但是只有一个缺点：不熟悉操作的人上手非常困难，需要人类记住大量的指令和规则，才可以有效的告诉机器它想要做什么——就像笑话里一样：“问：如何生成一个随机的字符串？答：让新手退出VIM”。

```
iTunes — -bash — 77x16
[CNxlewang:~ xlewang$ cd Music/
[CNxlewang:Music xlewang$ ls
iTunes
[CNxlewang:Music xlewang$ cd iTunes/
[CNxlewang:iTunes xlewang$ ls
Album Artwork                iTunes Library.itl
Previous iTunes Libraries      iTunes Media
iTunes Library Extras.itdb     sentinel
iTunes Library Genius.itdb
[CNxlewang:iTunes xlewang$ mkdir test
[CNxlewang:iTunes xlewang$ ls
Album Artwork                iTunes Library.itl
Previous iTunes Libraries      iTunes Media
iTunes Library Extras.itdb     sentinel
iTunes Library Genius.itdb     test
CNxlewang:iTunes xlewang$
```

直观的，既然“以机器的交流方式告诉机器想要做什么”这件事情给人类带来了很差的用户体验，那我们可以让机器提供可能的选项来让人类选择。所以，人类用了几十年，把交互系统升级成了图形化交互GUI。大家今天看到的桌面系统就是特别典型的一个体现。包括后来的触摸屏幕和智能手机的发展，其实都是图形化交互的不同表现。



现在一切都好了么？并没有。虽然机器可以瞬间呈现大量的信息，但是人类在同一时刻可能注意到的信息极为有限。心理学研究发现，人类的注意广度其实只有5-9个对象。想象一下上面那张图，如果我在桌面上放100个应用程序呢？1000个呢？随着数据量的发展。如何在大量的信息中，迅速呈现出有效的信息呢？

搜索系统，或者再具体一点，推荐系统，承担起了在选项过多的时候，给用户尽可能高效率的提供想要的信息的任务。如果我们做好了智能搜索，我们就能做好智能交互。本质上，他们都是一样的：在浩瀚的已知数据里，基于一定模型和经验，总结出用户最想要的答案并及时的呈现出来。我问Google一个问题，Google将我想要的答案排在第一个位置返回给我，谁又能说这不是对话式交互呢？


Google

weather of toronto

All News Maps Images Videos More Settings Tools

About 59,600,000 results (0.78 seconds)

Toronto, ON, Canada
Wednesday 1:00 AM
Light Thunderstorms and Rain









 **13** °C | °F

Precipitation: 64%
Humidity: 93%
Wind: 26 km/h

Temperature Precipitation Wind

13 12 14 15 17 16 17 16

2 AM 5 AM 8 AM 11 AM 2 PM 5 PM 8 PM 11 PM

| | | | | | | | |
|---|---|---|---|---|---|--|---|
| Wed | Thu | Fri | Sat | Sun | Mon | Tue | Wed |
|  |  |  |  |  |  |  |  |
| 17° 9° | 18° 11° | 18° 11° | 16° 11° | 14° 12° | 16° 12° | 17° 11° | 18° 12° |

More on weather.com Feedback

Toronto, Ontario 7 Day Weather Forecast - The Weather Network
<https://www.theweathernetwork.com/ca/weather/ontario/toronto> ▼
Find the most current and reliable 7 day weather forecasts, storm alerts, reports and information for Toronto, ON, CA with The Weather Network.

Hourly
Find the most current and reliable hourly weather forecasts, storm ...

36 Hours
Find the most current and reliable 36 hour weather forecasts ...

14 Day Trend
... current and reliable 14 day weather
... Toronto, ON, CA with ...

Toronto, ON
Conditions are favourable for the development of severe ...

当然，我们希望的对话式UI不仅是一问一答，我们希望他有自己的知识数据库，希望它保有对上下文的记忆和理解，希望它具有逻辑推演能力，甚至，颇有争议的，希望它具有一定的感情色彩。

所以，我们有了今天的Conversational UI，对话式交互只是一个壳子。其中的本质是智能和定制化服务，在一段时间的训练之后，你拿起电话拨给银行，应答的智能客服和人类的交互方式是一样的。抛开繁琐的从1按到9的决策路径，直接告诉他你要做什么，银行会直接给你提供最符合你需求的服务。而完成这个任务，我们主要有两条路可以走，一条是专家系统，这里也会给大家介绍几个网络上的引擎，争取在五分钟内让大家学会建立一个属于自己的智能客服系统。而另外一条，则是智能问答系统，需要一点机器学习和深度学习的知识——教机器理解规则，比教机器规则，要有趣的多。

输入和输出

前面都在讲输入，就是机器如何理解人类的指令。是因为输出这个问题，已经被解决了很久了。文本、图像和语音三大交流方式中，应该说语音被解决的最晚，但是20年前的

技术就已经足够和人类进行交流了，虽然我们还是很容易的听出来语料是不是电子合成，但是这一点音色上的损失并不影响我们交流的目的。

而语音到文本的识别便要复杂得多。这类工作确切来说始于1952年。从读识数字从1到0，然后把数字的声音谱线打出来，识别说的是哪个数字开始。这个模型虽然达到了98%的精度，但是其实并不具有通用性：数据源空间和目标空间都实在是太小了。

我们都知道当下最著名或者说最好用的语音识别模型是深度学习模型。但是在此之前呢？举一个典型例子：开复老师的博士论文，隐马尔科夫模型，大约三十年前发表，如下图所示：

A COMPOSITE FINITE-STATE NETWORK

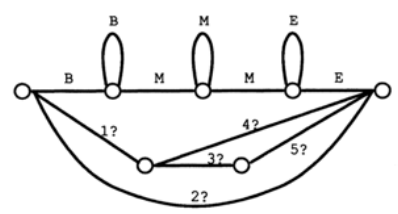


Figure 5-7: The HMM topology used in SPHINX, with different output pdf labelings on the lower transitions for different phones.

| 1 | 2 | 3 | 4 | 5 | Phones |
|---|---|---|---|---|---|
| B | B | B | B | E | /ae/, /eh/, /ah/, /aa/, /ao/, /uw/, /aw/, /ay/, /ey/, /ow/, /oy/, /l/, /en/, /er/, /m/, /n/, /ng/, /f/, /sh/, /v/, /z/, /sil/ |
| B | B | M | E | E | /ih/, /iy/, /uh/, /ax/, /ix/, /r/, /w/, /y/, /ch/, /jh/, /dx/ |
| E | E | E | E | E | /b/, /d/, /dh/, /g/, /k/, /p/, /t/, /s/, /th/, /hh/, /ts/ |

Table 5-2: Lower transition labels assigned for each phone using the HMM in Figure 5-7.

Lee, Kai-Fu. Automatic speech recognition: the development of the SPHINX system. Vol. 62. Springer Science & Business Media, 1988. @1988

简单说就是一个时间序列模型。有时间状态，隐藏状态，然后有观测状态。好像我有两个色子，一个六面体色子，从1到6，一个四面体的，从1到4，那我扔一段极端一点的序列 11112222444111166666666，大家觉得哪一段是四面体色子哪一段是六面体色子呢？听到一个语音，我想知道后面隐藏起来的那句话，原理也是和扔色子一样的：根据观测到的状态来推理后面隐藏的状态。这类概率模型的效果相当不错，以至于今天还有许多人用。

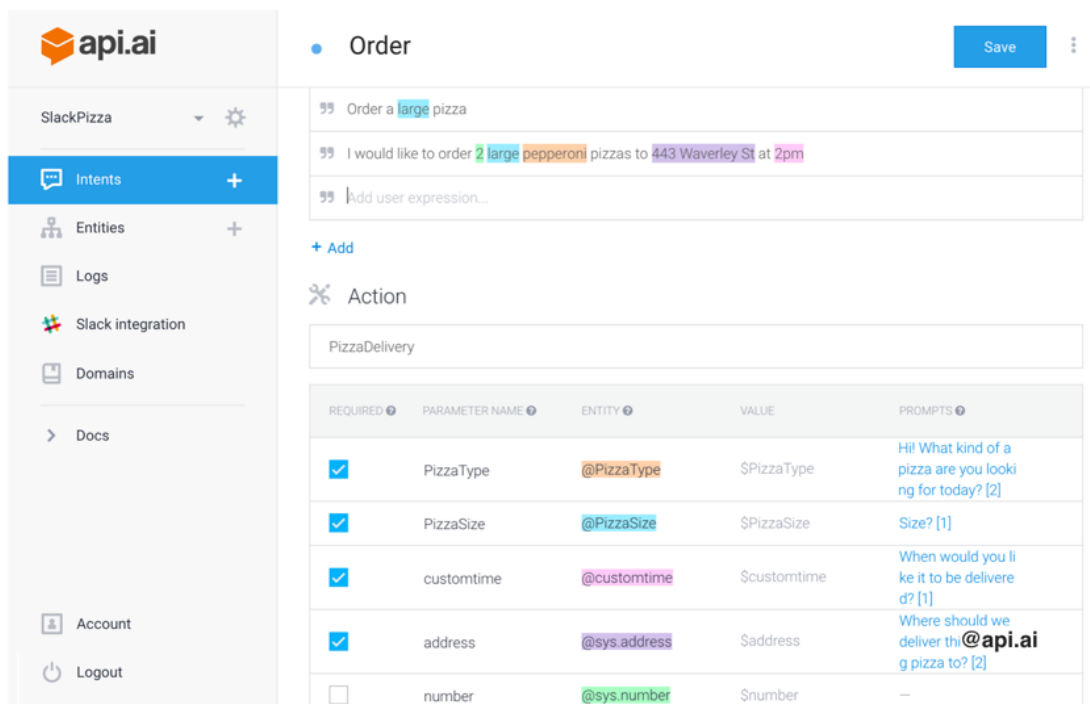
chatbot api

按照人工干预的多少，推理引擎的实现大致可以分为两类。一类是人工定义规则，一类是机器从数据里面自动学习规则。对于前者，我们都知道wit.ai和api.ai这两个著名的chatbot开放api, 分属于Facebook和Google两大巨头。先来看一下实现的效果（来源于<https://www.themarketingtechnologist.co/api-ai-vs-wit-ai/>）：

| Phrase | Api.ai | Wit.ai |
|----------------------------------|-----------------|------------------|
| I would like to order pizza | o | o |
| I want to order pizza | + | o |
| I am down for some pizza | + | o |
| I'd love some pizza | + | + |
| Medium | o | o |
| I'd like a small one | o | + |
| A large one, please | + | + |
| Get me an XXL | + | + |
| BBQ Chicken | o | o |
| I'll have a pepperoni, thank you | o | + |
| Do me a margherita, please | o | + |
| Quattro formaggi, would you | + | + |
| Emmasingel 25, 5611 AZ | o | o |
| Emmasingel 25, Eindhoven | + | + |
| 51.4402391, 5.47564740 | + | + |
| 5611 AZ 25 | + | + |
| Total | 9 points | 10 points |

这里的+表示得分，机器准确的理解了人类的意图。o表示不得分，机器并没有理解。我们可以看到，其实表现并没有想象中的那么好，一些很简单的案例‘i would like to order pizza’都没有得分，实在是离普通人类的智能还有些距离。

那么背后的逻辑是怎样的呢？可以参见下图：我们会定义不同的类型和变量，然后把他们和相关的值与回答链接起来。从而在和用户进行交互的时候，能够按照已知的（人类定义的）规则来存贮相应的值，并调用相应的方法。



可能英文大家会觉得毕竟读起来比较慢，这里介绍一个中文版api.ai——yige.ai. 并不是广告，我了解这个平台还得益于我的朋友——有一天他跑来跟我说：天寿啦！你知道吗，有个相亲网站，拿人工智能代替女性用户和人聊天！后来官方辟谣，说是协助不会聊天的人跟女生聊天。但是yige.ai作为一个相亲网站提供的api，在新手入门方面的友善程度，实在是我见过中数一数二的好。

具体参见：<http://docs.yige.ai/%E6%96%B0%E6%89%8B%E5%85%A5%E9%97%A8.html>



但是也正如图中所示，我们依旧需要人工定义很多事情包括词库，场景，规则，动作，参数等等。在买鞋这样一个小的场景和确定范围的交互期待里面，这样做还是可以为大部分人群所接受的。毕竟简单而直观，精准的实现了了“五分钟制作属于自己的chatbot”这一点，更不需要强大的计算资源和数据量。但我们并不太可能在这样的系统里面，得到定义好的域以外的知识。如果我们的时间和人力足够多的话，能够有专门的一些领域专家来完善这个提问库，将会使得搜索的精度非常高。因为所有可能的提问都已经有了专业的答案。但是，当场景复杂之后，这样做的工作量就会是很大的压力了。

所以，我们需要深度学习。

深度学习想要达到一个好的表现，需要有两个前提。一个是足量的计算资源，一个是大量的数据。

计算资源不用说，如果没有GPU，图片/语音这种非结构化的原始数据训练的时间基本需要以周来作单位计算。

数据集设计

关于大数据，一个很常见的问题就是，多大才算大，学术一点的说法是：大到包含区分目标值所需要的所有特征就可以了——我们都知道在实践中，这句话基本属于废话。那么换句话说吧，一般来说训练一个语音识别的模型，数据是以千小时为单位计算的。

数据来源：<http://www.cs.toronto.edu/~gdahl/papers/dbnVoiceSearchICASSP2011.pdf>

GitChat

4.1. Description of Dataset and GMM-HMM Baselines

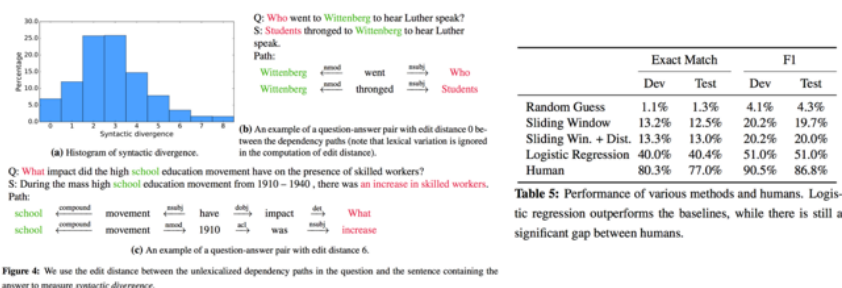
The Bing mobile voice search application allows users to do US-wide location and business lookup from their mobile phones via voice. This is a challenging task since the dataset contains all kinds of variations: noise, music, side-speech, accents, sloppy pronunciation, hesitation, repetition, interruption, and different audio channels. The dataset was split into a 24-hour (32,057 utterance) mini training set, which was a subset of the full 2100-hour (3M utterance) training set, a 6.5-hour (8,777 utterance) development set, and a 9.5-hour (12,758 utterance) test set. We ran all experiments on the mini training set in this study. To facilitate performance comparisons with the work in [13], which uses the same dataset and task, we used the public lexicon from Carnegie Mellon University. The language model (LM) used in the evaluation contains a 65K word vocabulary, 3.2 million word bi-grams, and 1.5 million word tri-grams. Performance on this task was evaluated using sentence accuracy (SA) instead of word accuracy for a variety of reasons. First, in order to compare our results with [13], we would need to compute sentence accuracy. Second, the average sentence length is 2.1 tokens, so sentence accuracy is highly correlated with the word accuracy. Third, the users care most about whether they can find the business or location they seek in the fewest attempts. They typically will repeat the whole sentence if one of the words is mis-recognized. Fourth, there is significant inconsistency in spelling that makes using sentence accuracy more convenient.

而且很抱歉的是，很多商业公司的数据集基本是不公开的。那么对于小型的创业公司和自由研究者，数据从哪里来呢？笔者整理了一些可以用来做自然语言处理和智能问答的公开数据集，这里由于篇幅和主题所限，就不展开讲了。改天会专门开主题介绍免费可用的公开数据，以及在公开数据集上所得到的模型应该如何迁移到自己的问题域当中来。

KNOWLEDGE DATA SET

SQuAD: 100,000+ Questions for Machine Comprehension of Text

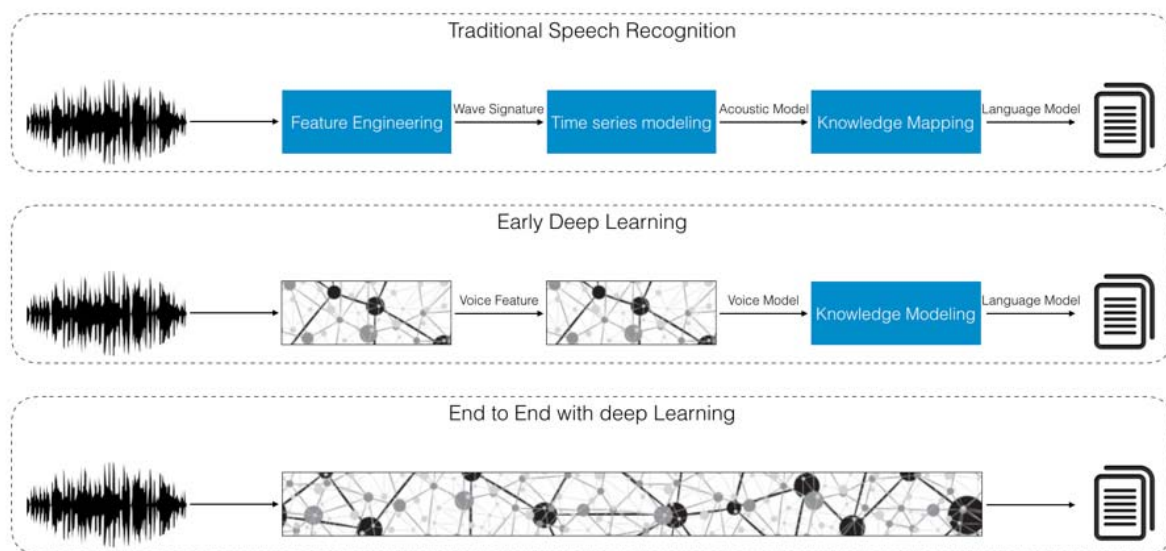
Pranav Rajpurkar and Jian Zhang and Konstantin Lopyrev and Percy Liang
{pranavs, zjian, klopyrev, pliang}@cs.stanford.edu
Computer Science Department
Stanford University



这里用斯坦福大学的著名问答数据集作为例子。我们可能在高中时代都做过阅读理解，一篇文章带有几个问题，答案来自于文章的信息。那么有了这样一个数据集，我们能做的事情是什么呢？这样一个数据集所训练出的模型可以解决什么样的问题？在各个问题中，人类的表现和机器的表现有什么样的差异？为什么？

深度学习 GitChat

好的，现在数据有了，计算资源有了，模型从哪里来呢？一个简单的进化路径如图所示。人工的干预随着技术的进步越来越少，直观一点，在图像识别中，神经网络每层的权重实际上学习到了图像的高级特征；越高层的神经网络，成分越具体。在第一层可能会把所有像素里面的点提出来，第二层可能是线，第三层可能是小的色泡或者面，第四层可能嘴巴，再往下可能是更复杂的人脸特征。神经网络和人脑一样，将原始信号经过逐层的处理，最终从部分到整体抽象为我们感知的物体。这是一个从图像到物体的感知过程，或者说是一个图像到标签列表的映射模型。



seq2seq

语音转文本或者问题到答案，也是一样的，可以用sequence2sequence作为学习的模型设计。前面说到的api.ai也好，yige.ai也好，规则和变量都是倾向于人工定义的。机器会对未经定义的语法规则给出一些通用的支持，但是正如我们看到的，一旦遇到定义域之外的交互场景，表现就很难尽如人意。

而在端到端的识别中，就如同上图中最后一行所展示的那样，我们不关心所有的语法和语义规则，所有的输入直接定向为问题，所有的输出直接是答案。当数据足够多，我们就可以做到端到端的识别，而不受人工定义的语义规则的干扰。这件事情，既是好事情，也是坏事情。基于人工规则的机器永远都不可能超过人类的表现，但是纯基于数据的机器学习模型，却可以打败人类——这点在AlphaGo的所向披靡之中，已经被证实过了。

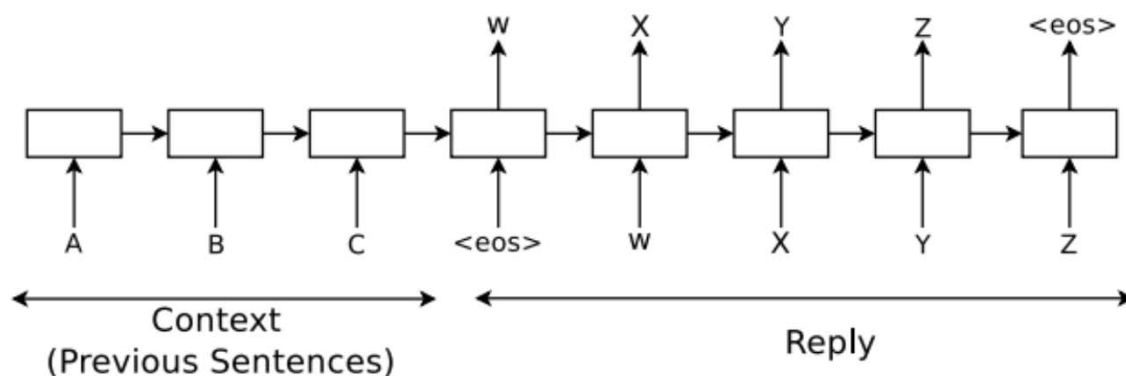


Figure 1. Using the *seq2seq* framework for modeling conversations.

如同图示，seq2seq的模型可以基于Sutskever在2014年发表于NIPS的一篇文章设计（<http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>），模型用recurrent neural network每次读入一个token作为输入，并预测应答的token。我们假设第一个人说了ABC，而第二个人回答了WXYZ，那么模型将会建立一个从ABC到WXYZ的映射。模型的隐变量，我们可以叫他“thought vector”，表明在这里机器理解了这个ABC的想法，或者说概念。这个模型在简化程度和通用程度上都是极好的，后面的实验也证明了这一点。

开源实现

相信通过前面的介绍，大家对于对话式交互系统，以及现有的api都有了初步了解，那么对于剩下一部分想要自己实现模型的人类，感谢github和arxiv，我们在源代码和原理级别都可以知道当今最聪明的那批人在做什么。

相对成熟的开源框架我推荐：<https://github.com/Conchylcultor/DeepQA>。

文章参见：<https://arxiv.org/pdf/1506.05869.pdf>。

和Google一直的风格相符，整个代码都是在TensorFlow和python3.5上实现，支持各种开源数据库以及定制化对话数据库，甚至拥有本地的web界面。通过TensorBoard我们也可以轻松监测系统的表现，虽然在部分对话的表现上差强人意，但是对入门者实在是再友好不过。

篇幅所限，这里不再啰嗦，祝大家玩的开心!

Reference

见文中链接。