如何做一款聊天机器人

原创 2016年07月28日 16:49:28 标签：语义 / 机器人 / 深度学习 / 大数据 / 爬虫

如何做一款聊天机器人

目录

前言

以此开题，并不是一定要做一款对话机器人，只是做事一定要以目标为驱动，目前所要研究的语义方向是一个太大的方向，是一个让人完全摸不到头脑的方向。因此，选取其中一个分支作为切入点，开始我的认知过程。在这个过程中，我会不断更新我的认识，以聊天机器人开篇，结尾处也许会面目全非！

什么是我认为的对话机器人

关于这个问题？相信大家都已经看过很多了。之前的小i、小白、图灵机器人、微软小冰、Cortana、HUBOT，google now，amazon echo 等。到底这些都是个什么鬼？大家可以简单了解下，更详细的请自行google。上面的几个机器人是我听过的，应该具有一定的代表性。

在google完这几个鬼之后，我发现我们需要的其实不仅仅是一款对话机器人，更确切的说，应该是个机器人助手。这个助手，应该能够跟人交流，这个交流不仅仅是你问我答，还可能有你说我做，你困或我解释等

可能有你说我做，你回忆我解释等。

给大家个直观的印象，贴一个Amazon echo的链接（如果不能访问，请翻墙）：

* https://www.youtube.com/watch?v=KkOCeAtKHIc 一个关于echo的视频，虽然并没有上下文语义，但是无论从立体声效果和应答的情感都是相当完美的。
* https://www.youtube.com/watch?v=24Hz9qjTDfw Echo dot是echo的迷你版
* https://www.youtube.com/watch?v=nVEEbKzaZFQ Echo tap 带有蓝牙和wifi功能，可以利用语音控制设备

总结下来，其实我们更需要的是一个语音助手。

语音助手

通过上面的得出的结论，对我们想要的东西，应该有个模糊的想象，我觉得就是那个样子，沿着这个思路，继续其明确我们的目标与实现途径。

如何去做一个像上述提到的那样的东西

简单的流程大约如下：

用户输入一段话（不一定只是单词）->后端语义引擎对用户输入的语句进行语义解析->推断用户最可能的意图->调用对应的知识库、应用、计算引擎->返回结果给用户。

实现方式有下面这么几种：

* ### 最初级的实现方法：关键词匹配（个人觉得我们的第一版实现可以采用这种方式）

> 建一个关键词词库，对用户输入的语句进行关键词匹配，然后调用对应的知识库。

此种方式入门门槛很低，基本上是个程序员都能实现，例如现在微信公众平台的智能回复、诸多网站的敏感词过滤就是此类。

但此种方式存在诸多问题，例如：

1. 由于是关键词匹配，如果用户输入的语句中出现多个关键词，此时由于涉及关键词权重（与知识库的关键词对比）等等问题，此时关键词匹配的方法就不擅长了
2. 不存在对用户输入语句语义的理解，导致会出现答非所问的现象。当然在产品上对回答不上的问题就采用卖萌的方式来规避掉。
3. 基本上无自学习能力，规则只能完全由人工维护，且规则基本是固定死的。
4. 性能、扩展性较差。还是上面的一句话中包含多个关键词的例子，采用普通程序语言来做关键词匹配，性能奇差。即便采用一些文本处理的算法来做（例如Double-array trie tre e），也很难满足大规模场景需求。

- 稍微高级点的实现方法：基于搜索引擎、文本挖掘、

自然语言处理（NLP）等技术来实现

相对于1的关键词匹配，此种实现方法要解决的核心的问题可以大致理解为：根据一段短文本（例如用户问的一句话）的语义，推测出用户最可能的意图，然后从海量知识库内容中找出相似度最高的结果。

具体技术实现就不细说了。举一个很粗糙的例子来简单说一下此种实现方法处理的思路（不严谨，只是为了说明思路）。

假如用户问：北京后天的温度是多少度?

如果采用纯搜索引擎的思路（基于文本挖掘、NLP的思路不尽相同，但可参考此思路），此时实际流程上分成几步处理：

1. 对输入语句分词，得到北京、后天、温度3个关键词。分词时候利用了预先建好的行业词库，“北京”符合预先建好的城市库、“后天”符合日期库、“温度”符合气象库
2. 将上述分词结果与规则库按照一定算法做匹配，得出匹配度最高的规则。假定在规则库中有一条天气的规则：城市库+日期库+气象库，从而大致可以推测用户可能想问某个地方某天的天气。
3. 对语义做具体解析，知道城市是北京，日期是后天，要获取的知识是天气预报
4. 调用第三方的天气接口，例如中国天气网-专业天气预报、气象服务门户 的数据
5. 将结果返回给用户

以上例子其实很粗糙，实际上还有诸多问题没提到：语义上下文、语义规则的优先级等等。

例如用户上一句问：北京后天的温度是多少度？下一句问：后天的空气质量呢？这里实际上还涉及语义上下文、用户历史喜好数据等等诸多问题。

此种处理方法存在的最大问题：规则库还主要依赖于人工的建立，虽然有一定的学习能力，但自我学习能力还是较弱。可以借助一些训练算法来完善规则，但效果并不是很好。而这也是目前流行的深度挖掘技术所擅长的。

- 当下时髦且高级的玩法：基于深度挖掘、大数据技术来实现

这种做法，要基于的技术就比较多了，总结为以下架构（盗图，来源已在参考帖子中注明）：

解释一下上述架构图

- ### 存储层
  对于这一层，个人认为就是互联网上或是本地的一切能够获取到数字资源（网页、视频等等等等），对于一些受限的资源（如QQ聊天记录等），也可以通过一定的方式获取到。

- ## 数字聚合层

  这一层的存在，其实是将互联网上杂乱无章的数据，进行各简单的分类，可能会用到一下三种方式：

  1. 人工维护录入数据（不做细说）
  2. 第三方开放平台接口数据
     1. 通俗的讲，所有你在网上注册的使用的，你以为是免费的东西，都能提供一种数据接入的方式，你的各种信息都被平台获取。当然，这只是一种方式，其他还有很多方式，请自行google。
     2. 再举个例子，现在我用的是搜狗输入法，如果你真的是它想免费给你提供输入法，那你就太天真了，too young, too naive

3. 垂直爬虫爬取数据

    1. 所谓垂直爬虫，通俗的讲，可以认为是针对某一领域或行业的爬虫。网上的数据毕竟是错综复杂的，用户所需获取的信息是需要有针对性的。比如，在垂直搜索的索引建立之前，我们需要到垂直网站上抓取资源并做一定的处理。垂直搜索与通用搜索不同之处在于，通用搜索不需要理会网站哪些资源是需要的，哪些是不需要的，一并抓取并将其文本部分做索引。而垂直搜索里，我们的目标网站往往在某一领域具有其专业性，其整体网站的结构相当规范(否则用户体验也是个灾难，想想东一篇文章西一篇文章基本没人会喜欢)，并且垂直搜索往往只需要其中一部分具有垂直性的资源，所以垂直爬虫相比通用爬虫更加精确。
    2. 两个垂直爬虫简介的链接：

    http://www.oschina.net/question/163158_109450

    http://liangqingyu.com/page/category.html#细说垂直型网络爬虫

推荐几个数据获取的网站：

- http://www.datatang.com/freelimit 数据堂
- http://www.datamall.com/ 数据商城

数字聚合层的数据，其实还是一些原始数据，是下一步针对性抽取的前提。

- 数据挖掘层

这一层体系，基本上是在有行业数据的基础上，进一步的对兴趣点进行提炼。基本也分为三个方向：

1. 文本挖掘
    1. 从海量文本中提取出有用的信息。如，处理和文本的表示，词的关联性挖掘及分析，话题的挖掘和分析，观点挖掘和情感分析，基于文本的预测。如，根据一段话来判断它的情绪，看看有没有反动言论等，这个都算是其中的一种。
    2. 给出几个链接，可以简单了解下：
        1. 文本挖掘和分析初步 http://www.jianshu.com/p/a98ac6847181
        2. 知识库：文本挖掘概述 http://udn.yyuap.com/doc/ae/919872.html
        3. 数据科学18：文本挖掘1 http://jackycode.github.io/blog/2014/06/18/text-mining1/
    3. 从狭义的角度看，文本挖掘是不做推理的，但现在挖掘技术总是和深度学习结合在一起的。
2. 协同过滤
    1. 协同过滤是利用集体智慧的一个典型方法。要理解什么是协同过滤 (Collab

orative Filtering, 简称 CF)，首先想一个简单的问题，如果你现在想看个电影，但你不知道具体看哪部，你会怎么做？大部分的人会问问周围的朋友，看看最近有什么好看的电影推荐，而我们一般更倾向于从口味比较类似的朋友那里得到推荐。这就是协同过滤的核心思想。换句话说，就是借鉴和你相关人群的观点来进行推荐，很好理解。

2. 你会发现微博或淘宝下面经常就会给你推荐小广告，这就是协同过滤。
3. 接着链接两篇帖子：
   https://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/
4. 对于我们来说，协同过滤的理念完全可以应用到产品中，帮用户进行各种需求的推荐。

3. 深度学习
   1. 其实，这一点是与其他技术相结合的。通过数据，按照各种算法进行学习训练，从而形成一套模型架构。利用训练好的模型，可以对未知的数据进行分析。这方面相关的东西太多，大家可以自行google。

数据挖掘层的输出，就是各种各样的知识库，是语义系统能够用到的最直接的东西。

- # 知识库层

这一层很好理解，其实更接近我们目前所能理解的东西。文本分析之后，去相应的知识库寻求问答。例如，对于一个机器人对话系统，你说一句话，语音转成文字之后，根据文字的分词、句法、语义分析结果，去对应的语言库中，寻求或自动生成最合理的应答。对于语音助手，那么先分析出，需要哪样的知识库，在去相应的知识库中寻求结果，或回一句话，或放个音乐，或开个空调，等等等等。介绍下知识库：

1. 通用知识库
2. 专用知识库
   1. 比如针对人机对话，音乐，地图等的库，都属于专用库。
3. 媒体库
   1. 你在google或百度用文字进行搜索，结果中有网页，也会有视频或图片，这就是从媒体库中进行的抽取
4. 社会化媒体库
   1. 简单介绍下，自行理解：
      http://wiki.mbalib.com/wiki/%E7%A4%BE%E4%BC%9A%E5%8C%96%E5%AA%92%E4%BD%93
5. 语义库和规则库
   1. 在我们的课题中，语义库和规则库，主要指文字到答复或是控制命令的转换规则。这种规则一部分是自己定义的，一部分可以利用深度学习，从大数据中进行训练学习得到的。

知识库，是我们要做这件事的重中之重，或购买第三方，或自己进行训练提取，难度呵

呵哒！！！

- 引擎层
  - 就个人看来就是个框架，没有知识库，它什么都干不了，大家概念一下就好。
- 解决方案层
  - 也不多说了，实际上就是你选择做个聊天机器人还是个语音助手之类的，巴拉巴拉！！！

如何将上述理论和实际结合呢？

这一步的假设是需求的资源都能获取到。

已做一个聊天机器人为例，讲述一下如何通过上述架构来实现：

1. 存储层，就是网上所有的数据，文字的、视频、音频都算。
2. 数据聚合层，在所有网上杂乱无章的数据中，其实我更需要的是QQ的聊天记录或是视频、音频的对话记录，需要有针对性的获取这部分数据。
3. 数据挖掘层，那么如何获取这些数据呢，可以通过购买或者是爬虫技术进行爬取。因为这些东西属于用户隐私，在使用爬虫进行爬取的时候可能需要一些黑客的技术融入其中，针对很多大的数据网站是有反爬机制的，还要想办法绕过这一关。在此实现的基础上，对对话内容进行NLU的分词、句法分析等操作，并将输出结果，作为深度神经网络的输入，进行训练。得到一套应答机制4。
4. 知识库层，利用3输出的结果，构建应答的语义和规则库
5. 引擎层， 这一部分包括语音识别（声音转文字），语义理解引擎（对文字进行分词和句法分析，将分析结果输入上面训练的神经网络，得到答复），语音合成（将输出的结果最终的读出来）。
6. 解决方案层， 构建一个聊天机器人的应用，包括UI界面和交互逻辑等。

结论

整个上面的过程，实际上是我一个门外汉，对整个我们要做的事情的一个理解的过程。

参考文献（帖子）

- 微软小冰智能聊天是如何实现的？ https://www.zhihu.com/question/23952075

- 垂直爬虫
  - http://www.oschina.net/question/163158_109450
  - http://liangqingyu.com/page/category.html#细说垂直型网络爬虫
- 文本挖掘
  - 文本挖掘和分析初步 http://www.jianshu.com/p/a98ac6847181
  - 知识库 : 文本挖掘概述 http://udn.yyuap.com/doc/ae/919872.html
  - 数据科学18：文本挖掘1 http://jackycode.github.io/blog/2014/06/18/text-mining1/
- 协同过滤
  - 探索推荐引擎内部的秘密，第 2 部分: 深入推荐引擎相关算法 - 协同过滤https://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/
- 语义理解
  - 地图中的语义理解 | 硬创公开课 http://weibo.com/ttarticle/p/show?id=2309351000223982624799503109
- 社会化媒体
  - http://wiki.mbalib.com/wiki/%E7%A4%BE%E4%BC%9A%E5%8C%96%E5%AA%92%E4%BD%93