

基于自然语言处理的应用自动生成系统 - OfficeCoder

邢加荣¹, 许崇杨², 孙铭超³, 罗远航⁴

山东大学软件学院^{1,2,3,4}

xingjiarong@mail.sdu.edu.cn¹ xuchongyang@mail.sdu.edu.cn²
sun.mc@outlook.com³ luoyuanhang@mail.sdu.edu.cn⁴

Abstract. 近年来, 随着互联网行业的飞速发展和移动智能终端的升级换代, 人们对手机应用的需求量越来越大。但是, 目前的软件行业依然采用传统的开发方式即编写程序进行开发, 这种方式既费时又费力, 同时它要求开发者必须具备足够的编程知识才能进行设计和开发。这就导致了软件定制的困难, 使得用户不能直接设计自己的软件。本文中, 我们提出了一种软件自动生成系统, 该系统利用自然语言处理技术, 将用户对想要生成的应用的描述转化为真正的应用, 从而不需要编程就能够设计自己的软件。加之, 我们利用了云计算强大的计算和存储能力, 在云端为用户准备了大量的素材库, 极大提高了应用生成的准确性。本系统依托于 Microsoft Word 2013, 以插件的形式安装在 Word 2013 上, 简化了程序的安装和使用, 同时借助了 Office 的用户群体, 扩大了市场占有率。在软件行业蓬勃发展的今天, 我们的系统提出了一种新的软件开发方案, 力图改变原有的开发方式, 使得每个人都能在不具备专业知识的前提下, 设计自己的软件, 实现软件定制...

Keywords: 自然语言处理, 云计算, HTML5, Office, 移动应用, XML

1 引言

随着互联网行业飞速发展, 移动终端的迅速普及, 移动软件成为了各行各业不可或缺的生产力工具, 同时也是国家“互联网 +”产业升级中的重要组成部分。目前无论是初创企业, 新型媒体, 甚至是个人都对移动应用软件有着或多或少的需求。而软件编写高成本周期长的特点无疑是与目前形势相矛盾的, 无论是对于希望产品迅速上线抢占市场的初创企业还是对于不希望承受巨大开发成本的有定制需求的个人, 传统的软件开发方式都无法满足他们的需求。因此我们力图去降低软件开发的门槛, 为各行各业有软件需求的人士提供制作自己定制软件的一套解决方案。

对于非专业人士, 我们认为通过自然语言描述和可视化界面拖拽的交互方式是最为自然的。但对于使用编程语言都难以编写的软件, 使用自然语言来编写软件自然更是难上加难。不过, 随着近年来计算机运算水平的提升, 人们在统计学习算法方面更加深入的研究, 自然语言处理技术也随之蓬勃发展, 自然语言处理已经广泛应用于语音助手, 计算广告学等领域, 其对语义的分析也达到了一定的高度。也因此我们敢于去试图让计算机理解人对软件需求的描述, 根据语义自动化的完成原本复杂的软件编写。我们针对于这个问题设计完成了我们的一整套系统, 包括为用户提供语言与可视交互的客户端, 对自然语言进行分析, 自动生成代码编译为软件包的服务器, 在我们的自然语言处理算法的驱动下一定程度上完成了最初设想的功能。

用户在使用我们系统的时候只需要输入对软件功能和界面比较详细的描述, 我们的系统会通过对这段文字的分析, 提取其中的关键词和语义信息, 尽可能的理解用户的意图, 根据解析结果将会从我们构建的代码库中选取合适的模块按照一定方式进行组合, 并赋予所需的功能, 最终返回给用户一个软件的雏形。之后用户可以继续使用客户端提供的可视化工具对软件细节进行进一步的调整和修改, 修改结束后软件会自动打包成安装包, 供用户下载安装。

我们考虑到目前在办公领域微软 Office 占有大量的份额, 几乎各行各业每个人在办公时都离不开微软提供的 Office 办公套件, 因此客户端我们采用了微软 Office 系列软件插件的形式。经过简单的插件安装, 每个用户都可以在非常熟悉的 Microsoft Word 中完成软件的定制。依托于经历了多年考验的 Office, 我们的软件稳定性得以保障, 同时得益于 Office 广大的用户群体, 我们软件的推广也会轻松很多。

服务端我们采用了微软提供的 Azure 云服务, 依托微软云强大的计算性能, 我们得以完成复杂的自然语言分析与代码自动生成。我们在充分利用计算资源的同时, 利用宽广的云存储构建了模块化的代码仓库。在客户端与服务端都采用微软的产品也使得我们可以与 Office 的 One Drive 交互, 为用户提供更为全面的服务。

我们希望通过我们的努力能打破人们对软件编写的认知, 通过自然语言的形式真正让普通人没有门槛的完成个人软件的定制, 让写一款软件变得触手可及。面对这个比较困难的问题, 我们在架构和算法等多个技术点进行了大胆的设计与创新, 最终取得了不错的效果。

2 相关工作

目前软件快速生成平台也有一些, 例如简网 app 工厂等, 但是他们普遍都存在套用模板, 灵活性不够等等情况。用这些平台来制作一款应用时, 不仅需要花费很多时间去寻找适合的模板, 而且很多模板的设计和请求出入比较大, 有时用户需要再手动调整很多内容, 有时则因为差距实在过大而难以使用。总之, 目前现有的快速生成平台完全没有使用到自然语言处理技术, 而仅仅停留在选择模板并加以更改的层面上。而我们通过使用自然语言分析可以从一段需求描述直接生成软件雏形, 用户需要什么样的软件只需要用自然语言描述出来, 我们的算法则会生成出比较接近用户所想的软件雏形, 也因此, 后续的调整过程会比较简单, 而相比寻找模板再进行修改更是要快速与方便很多。

目前常用的分词方法主要有 3 种: 基于词表的分词方法、基于统计模型的分词方法和基于统计方法与规则方法相结合的分词方法 [5]。3 种方法均有优点, 但也存在不足: 基于词表的分词方法效率高, 但对新词的识别能力不足 [6]; 基于规则的方法很难涵盖所有的语言现象 [5], 尤其对网络语料的处理能力非常有限; 基于统计模型的分词方法 [8] 重点在于解决自动分词的歧义分词问题, 但需要人工标注训练语料 [1], 且受训练语料领域的限制。ICTCLAS(Institute of Computing Technology, Chinese Lexical Analysis System) 是基于隐马尔科夫统计模型 (HMM, Hidden Markov Model)[9] 进行分词的广受好评的中文分词系统, ICTCLAS2002 版在国内 973 评测中综合第一名, 经过 15 年打造, ICTCLAS2015 版又增加了新词自动识别功能。因此我们在 ICTCLAS2015 基础上进行分词, 并对分词后的词语进行词性标注 [10], 通过词性将一些无意义的词语去掉, 对语料

进行净化。并根据我们软件生成系统的需求，向词典中添加一些新的词语，使得最终分析的结果更加精确。

句法分析是自然语言理解中一个至关重要的环节。目前的句法分析主要有两种方法：基于规则的方法和基于统计的方法。相比于传统的基于规则的方法，基于统计的分析方法具有更好的灵活性，在处理歧义等不确定性问题方面都有较好的效果。使得统计方法渐渐成为自然语言处理的主流 [7]。概率上下文无关文法 (PCFG) 是把统计方法引入到上下文无关语法规则系统而形成的语法规则系统。然而经典的 PCFG 实际上是建立在一些非常理想化的独立性假设的基础之上的，而这些假设并不符合实际，于是造成了 PCFG 的实际效果不理想。目前的概率上下文无关语法研究，主要集中在如何突破这些独立性假设上。通过逐步的放宽这些假设，分析的正确率可以得到很大的提高。

在自然语言处理系统中，我们实现了 ICTCLAS2015 的分词算法来获取用户原始语料，进行精确分词，然后通过基于贝叶斯模型的机器学习算法训练大量样本，达到精确匹配控件的目的。通过基于 PCFG 概率上下文无关的句法分析 [2] 算法构建优化句法树，分析语义逻辑，然后将结果通过基于谓词的句法树遍历模型算法来精确获取空间属性。最终各种机器学习算法、智能算法的应用与优化使得整个应用自动生成系统更加准确，可以准确的从用户的文字描述中获取到关键信息，能够最大程度的使得该系统生成出的应用能够满足用户的需求。

3 系统概要设计

3.1 综述

全过程软件自动生成技术是一种计算机辅助软件制作技术。它根据自然语言分析，将普通人所能理解的自然语言，经过逐步的转化最终生成出能够被我们所编写的软件生成引擎所识别的软件生成脚本。整个过程包括定性层和定量层两大部分，定性层确定了用户所提供的自然语言所包括的软件模板、软件控件、软件主题等，展开自然语言中所表示出的各个控件、模板之间的层次关系，以明确用户所表述的自然语言中各个部分所涉及到的具体控件、软件锁应当具有的行为、功能；定量层把生成的软件控件的位置数值化（坐标化），根据自然语言分析将控件的位置、颜色、大小等按照用户的要求拼装于最终生成的软件中。

3.2 客户端

客户端以 Microsoft Word 插件形式嵌入到 Microsoft Word 之中，成为 Microsoft Word 的一部分。用户在 Word 文字输入区域输入一段文字描述，在扩展的工具栏中使用功能键，即可进行相应操作：点击预览按钮获得系统自动生成的 App 的“所见即所得”的预览；在预览界面上选择控件，进行拖拽，缩放，调整属性等操作；进行添加删除控件；确认生成最终 App 等操作。

UI 设计 UI 设计整体采用扁平化风格设计，以和 Word 的风格相一致。插件的功能按键位于 Word 的工具栏，和 Word 整体风格相一致。智能手机预览，和 Word 文本输入区域左右布局，符合用户的心理。

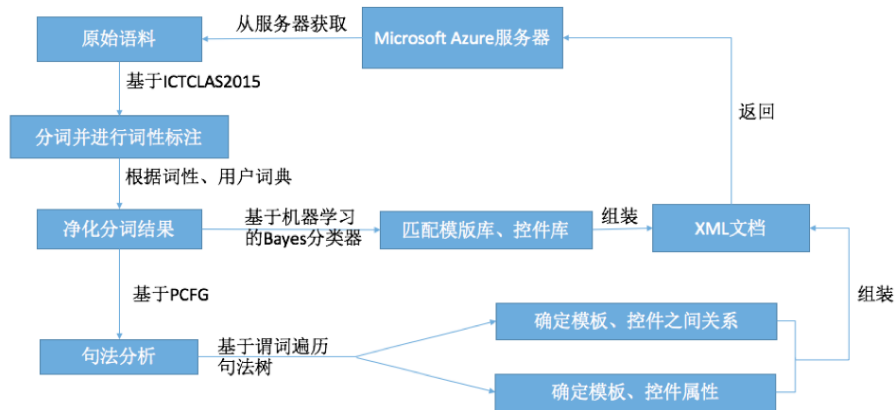


Fig. 1. 系统流程

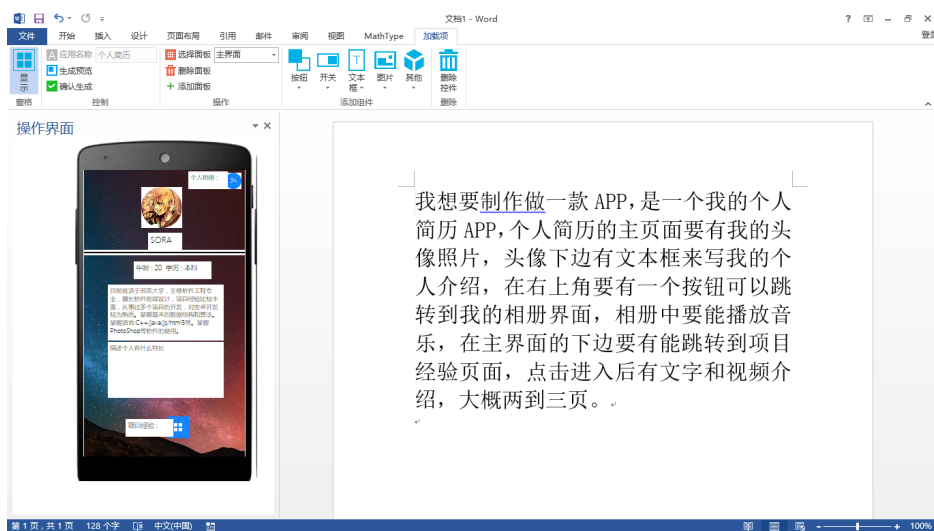


Fig. 2. 主界面设计

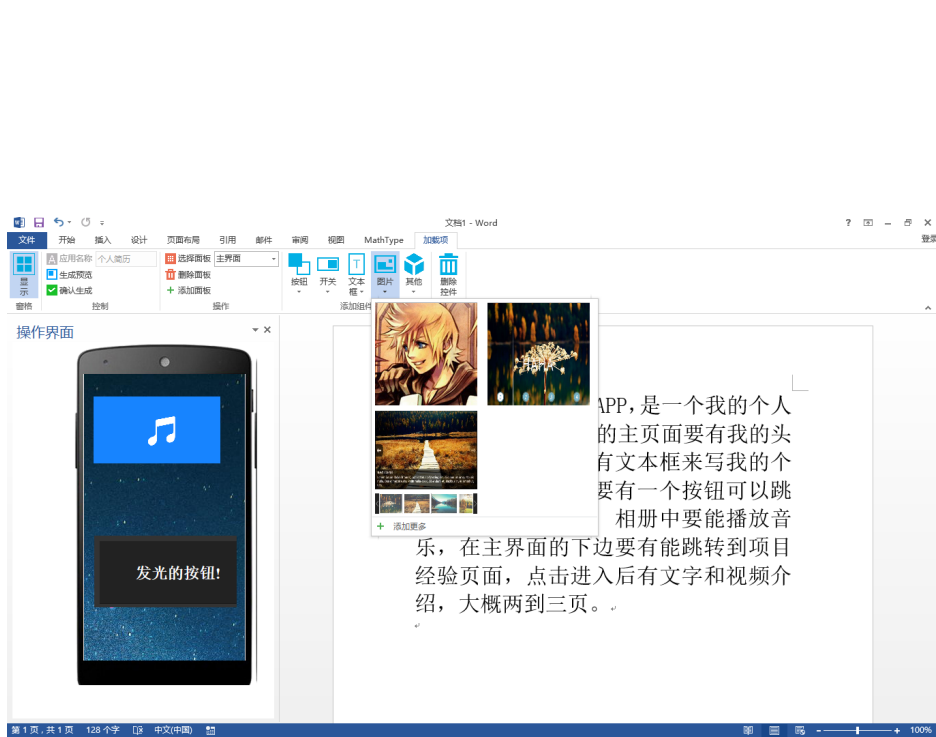


Fig. 3. 下拉功能菜单设计

XML 描述文件和 App 预览的相互转换 自定义 XML 文件规则, 定义一个 App 的机器描述, 定义 Application 标签, Page 标签, 各种控件标签, 控件标签含有在页面的位置信息, 属性信息等。客户端解析 XML 文件, 分析出 Page 层, 控件层对应的组件, 并根据解析出的属性信息初始化对应的 C 图形控件, 根据位置大小颜色内容等属性, 展示在可见即所得的侧栏中。XML 文件举例如下:

图形化操作的实现 为每一个控件进行封装, 将每一个 Page 中解析出的控件存储在一个队列之中, 并为 Page 实现一个鼠标事件监听, 在鼠标点击时确定选中的控件, 将控件移动至队列头部, 重新绘制队列, 对于选中的控件, 鼠标右击弹出修改属性的面板, 可以修改属性, 在选中的控件周围有重新调整大小的控制点, 点击后拖拽即可实现重新调正大小。

3.3 基于 MICROSOFT AZURE 的云服务器

时至今日，IT 服务已经成为任何商业运作的必备设施。云计算就是要将 IT 服务变得像用电一样简单。企业不需要担心电力来自哪家发电站、电线如何布设。同理，在云计算的帮助下，企业不再费心管理每台服务器用什么处理器、装什么操作系统或者数据库。在众多的云服务中，我们选择微软的 Microsoft Azure。

Azure 是一种灵活和支持互操作的平台，它可以被用来创建云中运行的应用或者通过基于云的特性来加强现有应用。它开放式的架构给开发者提供了 Web 应用、互联设备的应用、个人电脑、服务器、或者提供最优在线复杂解决方案的选择。Windows Azure 以云技术为核心，提供了软件 + 服务的计算方法。它是 Azure 服务平台的基础。Azure 能够将处于云端的开发者个人能力，同微软全球数据中心网络托管的服务，比如存储、计算和网络基础设施服务，紧密结合起来。

我们在 Azure 云上部署了我们的服务器。服务器分为计算和存储两部分。计算部分负责调用算法和组装编译打包成 APK，存储部分负责将收到的来自客户端的音乐、视频和图片等信息存储下来，这些信息主要用于生成用户的 APP。同时在我们还需要保存整个控件库，因为控件库是整个系统的核心，所有的 APP 都是由控件库中的空间经过组合和调整之后得到的。

工作流程 服务器作为处理核心负责调用各种资源。服务器从客户端获得原始的文本输入，然后经过自然语言分析，得到用户语义，接着将组装成 XML 格式的文件返回给用户进行调整，然后根据用户的调整调用控件进行拼装，最终将编译好的 APP 以二维码的形式发送给用户。详细的步骤如下：

- 1) 首先客户端将用户在 Word 中输入的用于描述自己想要的 APP 的文字使用网络发送到服务器
- 2) 服务器得到客户端传来的文字后，调用自然语言分析算法进行语言分析，将得到的结果组成一个 XML 的格式。然后将这个 XML 文件返回给客户端
- 3) 客户端可以通过界面上的预览窗口看到目前的 APP 界面内容，此时用户可以通过拖拽的方式进行调整，所有的调整都被记录在 XML 文件中
- 4) 用户调整完成后，XML 内容被发送到服务器。服务器根据最终的这份 XML 文件调用控件库中的控件，按照 XML 中的规则进行组装，组装成 HTML5 代码
- 5) 服务器调用安卓的编译打包程序，将组装好的 HTML5 代码编译成安卓的 APK
- 6) 调用二维码生成算法，将 APK 的下载地址以二维码的形式返回给客户端，供客户端下载

客户端和服务端之间采用 HTTP 的 POST 方式进行通讯，相对于 SOCKET 的长连接通讯，POST 的方式更加的低消耗，也就是说服务器可以同时为更多的客户端进行服务。采用 XML 的格式进行数据传输而不是使用 JSON 或者字符串，提高了传输的效率，降低了解析的难度，是服务器和客户端处理起来更加方便。

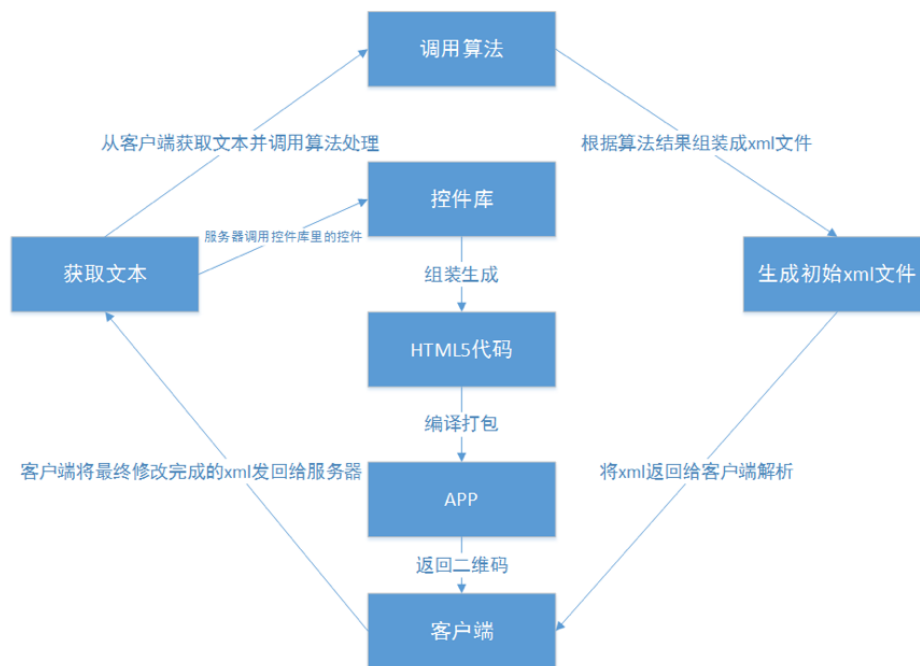


Fig. 6. 服务器端流程图

文件存储 在用户需要生成的 APP 中，有很多 APP 都需要播放视频或者音乐，这些视频或音乐有一些来自网络，有一些是由用户自己上传的。那些需要用户上传的视频就需要在服务器上存储下来，因为会有下面一种情况。用户自己上传视频，生成 APP，但是用了一段时间后用户换了新手机或者重新安装了系统，所以以前保存在手机上的视频就都不存在了，这时候如果服务器端没有存储用户上传的视频和音乐的话，APP 就不能正常的播放视频和音乐了。

我们采用 Azure 的云存储方案。Azure 文件存储是一种使用标准服务器消息块 (SMB) 协议在云中提供文件共享的服务。文件存储使用标准 SMB 2.1 或 SMB 3.0 协议为应用程序提供共享存储。Azure 虚拟机和云服务可通过装载的共享在应用程序组件之间共享文件数据，本地应用程序可通过文件存储 API 来访问共享中的文件数据。在 Azure 虚拟机或云服务中运行的应用程序可以装载文件存储共享以访问文件数据，就像桌面应用程序可以装载典型 SMB 共享一样。任意数量的 Azure 虚拟机或角色可以同时装载并访问文件存储共享。

文件存储包含以下组件：

- 存储帐户：对 Azure 存储服务的所有访问都要通过存储帐户来完成。有关存储帐户容量的详细信息，请参阅 Azure 存储空间可伸缩性和性能目标；
- 共享：文件存储共享是 Azure 中的 SMB 文件共享。所有目录和文件都必须在父共享中创建。一个帐户可以包含无限数量的共享，一个共享可以存储无限数量的文件，直达到文件共享的 5TB 总容量限制为止；
- 目录：可选的目录层次结构；

- 文件：共享中的文件, 文件大小最大可以为 1 TB;
- URL 格式：可使用以下示例 URL 寻址下图中的文件：
`http://samples.file.core.chinacloudapi.cn/logs/CustomLogs/Log1.txt`

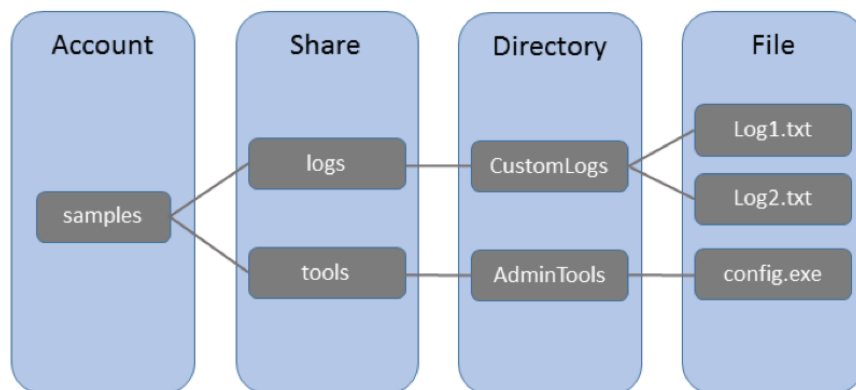


Fig. 7. URL 寻址

组装 xml 客户端和服务端之间需要传递要生成的 APP 之间的格式信息，如果将所有的资源连同信息一起传递，将会既费时又费力，所以我们采用不传递资源，只传递格式的方法。在众多的格式方式方案中，我们选择了 XML 方案。

XML 即可扩展标记语言 (Extensible Markup Language)，是一种平台无关的表示数据的方法。简单地说，使用 XML 创建的数据可以被任何应用程序在任何平台上读取。甚至可以通过手动编码来编辑和创建 XML 文档。其原因是，XML 与 HTML 一样，都是建立在相同的基于标记技术基础之上。

我们选择 XML 以下几个原因：

- 在真实的世界中，计算机系统和数据使用不兼容的格式来存储数据。XML 数据以纯文本格式进行存储，因此提供了一种独立于软件和硬件的数据存储方法。让创建不同应用程序可以共享的数据变得更加容易；
- 通过 XML，可以在不兼容的系统之间轻松地交换数据。对开发人员来说，其中一项最费时的挑战一直是在因特网上的不兼容系统之间交换数据。由于可以通过各种不兼容的应用程序来读取数据，以 XML 交换数据降低了这种复杂性；
- XML 数据以文本格式存储。这使得 XML 在不损失数据的情况下，更容易扩展或升级到新的操作系统、新应用程序或新的浏览器。由于可以通过各种不兼容的应用程序来读取数据，以 XML 交换数据降低了这种复杂性

编译打包 我们在服务器上最初生成的版本是 HTML5 格式的，这种格式主要用于网站，不能直接在用户的手机上进行安装，所以我们需要将 HTML5 的网页内

容打包成安卓系统的 APK。我们采用安卓的 ADT 程序进行打包，安卓的 ADT 程序可以直接将 HTML5 打包进安卓的应用程序中，从而使用 HTML5 写的程序可以运行在安卓平台上。因为我们的应用程序是用 HTML5 编写的，HTML5 具有跨平台的特性，所以使用 HTML5 一次编写的程序可用打包到安卓，IOS 和 Windows 平台上，这样就避免了多次的尴尬的弊端。

4 算法设计及优化

在该系统的算法设计以及优化的部分，我们首先采用了 ICTCLAS2015 算法来对用户的原始语料进行分词，然后通过实现和优化贝叶斯算法 [3] 来对分词后的语料进行分类和训练。同时，将语料通过基于 PCFG 概率上下文无关的句法分析 [2] 算法构建优化句法树，来分析语义逻辑，然后将结果通过基于谓词的句法树遍历模型 [4] 算法来精确获取空间属性，最终完成整个自然语言处理的过程，将分析结果应用于应用自动生成系统中的模板匹配部分。

自然语言处理 (NLP) 作为该应用自动生成系统的重要组成部分，影响着该系统自动生成出的应用的相关程度以及软件质量，是该系统的关键部分。该自然语言处理系统分为以下几个子系统：

- 自然语言分词子系统¹；
- 控件精确匹配子系统²；
- 句法分析子系统³；

在自然语言分词子系统¹中，我们使用 ICTCLAS2015 进行分词，并对分词后的词语进行词性标注，通过词性将一些无意义的词语去掉，对语料进行净化。并根据我们软件生成系统的需求，向词典中添加一些新的词语，使得最终分析的结果更加精确。对于控件精确匹配子系统²，我们运用机器学习的方法，建立起 Bayes 分类模型 [3]，通过训练大量的文本，来将文本中的特征文本与模板库、控件库建立起对应的关系，能够较精确的仅仅通过文字来确定用户所需要的软件模板以及控件。在句法分析子系统³的构建中，用户文档中的语言方式都存在一些结构和规则。语言学中句法分析的目标就是努力分离出这些语法结构。现在我们一般所说的语法实际上就是指描述词语排列的方法，而排列的方式或者基于词，或者基于词类。即使在最传统的语法形式中，语法也不仅仅意味着现行词语之间的简单顺序。PCFG 概率上下文无关文法 [2] 是最简单的一种表达递归潜入概念的概率模型。它给简单 CFG 上下文无关文法中的规则添加了一些概率，指明了不同重写规则的可能性大小。

4.1 自然语言分词算法

我们使用 ICTCLAS2015 进行分词，并对分词后的词语进行词性标注，通过词性将一些无意义的词语去掉，对语料进行净化。并根据我们软件生成系统的需求，向词典中添加一些新的词语，使得最终分析的结果更加精确。

ICTCLAS 2015 中文词法分析是中文信息处理的基础与关键。中国科学院计算技术研究所多年研究工作积累的基础上，研制出了汉语词法分析系统 ICTCLAS(Institute of Computing Technology, Chinese Lexical Analysis

System), 主要功能包括中文分词; 词性标注; 命名实体识别; 新词识别; 同时支持用户词典。先后精心打造五年, 内核升级 6 次, 目前已经升级到了 ICTCLAS3.0。ICTCLAS3.0 分词速度单机 996KB/s, 分词精度 98.45%, API 不超过 200KB, 各种词典数据压缩后不到 3M, 是当前世界上最好的汉语词法分析器。

算法优化 我们在 ICTCLAS2015 的基础上对分词后的词语进行词性标注, 通过词性将一些无意义的词语去掉, 对语料进行净化。并根据我们软件生成系统的需求, 向词典中添加一些新的词语, 使得最终分析的结果更加精确。

Algorithm 1 自然语言分词优化算法

Require: *String* 用户语料

Ensure: 分词结果

```
1: function ICTCLAS2015 分词器 (String)
2:   result  $\leftarrow$  null
3:   if String  $\neq$  null then
4:     分词
5:     词性标注
6:     语料净化 (String)
7:   end if
8:   return result
9: end function
10: function 语料净化 (String)
11:   if String 相关 then
12:     将 String 加入词典
13:   else
14:     去掉无意义词汇
15:   end if
16: end function
```

4.2 贝叶斯分类算法

在文本分类中, 假设我们有一个文档 $d \in X$, X 是文档向量空间 (document space), 和一个固定的类集合 $C = c_1, c_2, \dots, c_j$, 类别又称为标签。显然, 文档向量空间是一个高维度空间。我们把一堆打了标签的文档集合 $\langle d, c \rangle$ 作为训练样本, $\langle d, c \rangle \in X \times C$ 。我们期望用某种训练算法, 训练出一个函数 γ , 能够将文档映射到某一个类别: $\gamma: X \rightarrow C$

Algorithm 2 贝叶斯分类算法

Require: C 类别集合, D 用于训练的文档集合

Ensure: 分类结果

```
1: function TRAIN( $C, D$ )
2:    $V \leftarrow \text{ExtractVocabulary}(D)$ 
3:    $N \leftarrow \text{CountTokens}(D)$ 
4:   if  $c \in C$  then
5:      $N_c \leftarrow \text{CountTokensInClass}(D, c)$ 
6:      $\text{prior}[c] \leftarrow N_c/N$ 
7:      $\text{textc} \leftarrow \text{ConcatenateTextOfAllDocs}(D, c)$ 
8:     if  $t \in V$  then
9:        $T_{ct} \leftarrow \text{CountTokenofTerms}(\text{textc}, t)$ 
10:       $\text{condprob}[t][c]$ 
11:    end if
12:  end if
13: end function
```

4.3 句法分析算法

本系统采用 PCFG 建立起句法树, 创新的采用基于谓词的句法分析树遍历方法, 优化传统的句法分析, 在软件的自动生成过程中更加精准的把握住软件中存在的逻辑关系以及用户的目的, 使得生成出的软件最大限度的契合用户的想法。

PCFG 概率上下文无关文法 概率上下文无关语法 (Probabilistic Context-free Grammar, 简称 PCFG) 又叫做随机上下文无关语法 (Stochastic Context-free Grammar, 简称 SCFG)。这种语法是由 Booth(1969) 最早提出来的。上下文无关语法可以定义为四元组 $\{N, \Sigma, P, S\}$ 。而概率上下文无关语法则在每一个重写规则 $A \rightarrow \beta$ 上增加一个条件概率 $A \rightarrow \beta[p]$ 这样, 上下文无关语法就可定义为一个五元组 $\{N, \Sigma, P, S, D\}$, 其中 D 是给每一个规则指派概率 P 的函数。这个函数表示对于某个非终极符号 A 重写为符号串 B 时的概率 P 。这个规则可写为: $P(A \rightarrow \beta)$ 。或者写为: $P(A \rightarrow \beta|A)$ 。从一个非终极符号 A 重写为 B 时, 应该考虑一切可能的情况, 并且其概率之和应该等于 1。

基于谓词的句法分析树遍历 对于已经构建好的句法树我们创新的采取了基于谓词的句法分析树遍历方法, 传统的后序遍历句法树的方法不能够满足该系统中语义分析的要求。在我们的获得的用户语料中, 在语法中最关键的是谓语部分, 它表示了在该用户语料中, 用户最主要的目的, 因此我们采用基于谓词来遍历句法树, 最大限度的保留了用户的需求目的, 能够更好的更精准的对我们所建立的模板库进行匹配。

5 实验与分析

5.1 使用 OfficeCoder 生成 App

打开 Microsoft Office Word, 输入一段描述软件需求的文字, 单击加载项, 即可看见 OfficeCoder 面板

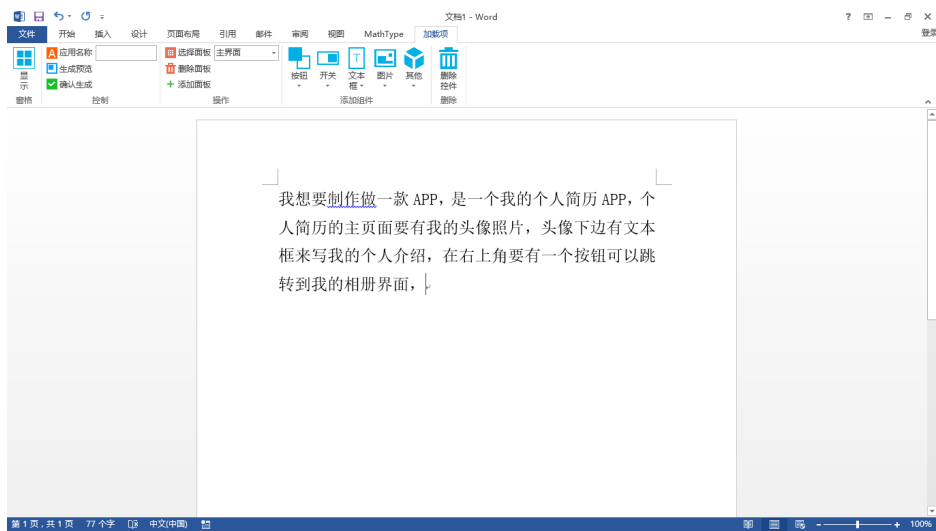


Fig. 8. OfficeCoder 面板

在“应用名称”一栏输入您期望生成的应用的名称，单击“生成预览”即可进入如下界面



Fig. 9. 预览生成效果

在生成的程序预览中，右击控件即可设置相关控件的属性，如：右击文本框，修改文本框的属性



Fig. 10. 修改相关控件属性

如果对生成的模板、控件不满意，可以对控件等进行拖动，更改位置、大小等。

单击 OfficeCoder 面板中的“添加面板”即可添加新的页面

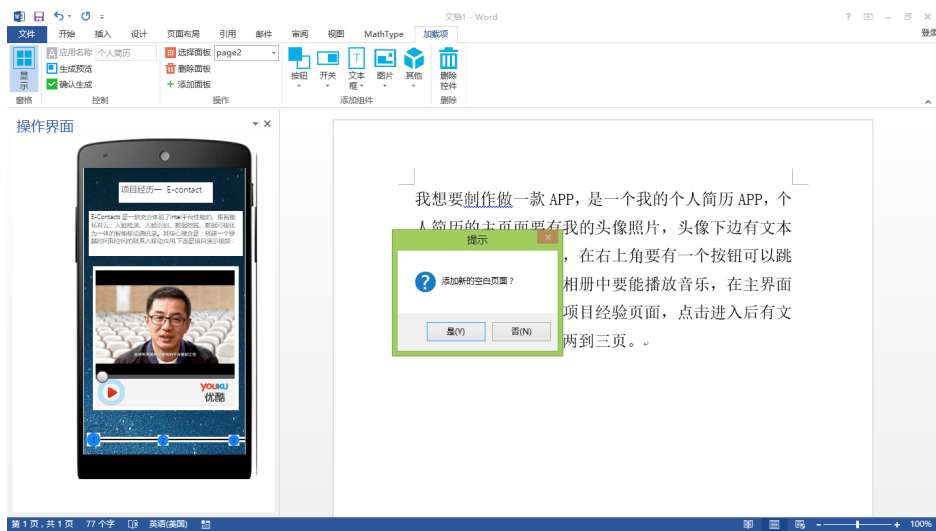


Fig. 11. 添加新页面

单击“按钮”“开关”“文本框”等控件，即可在页面中添加相应控件

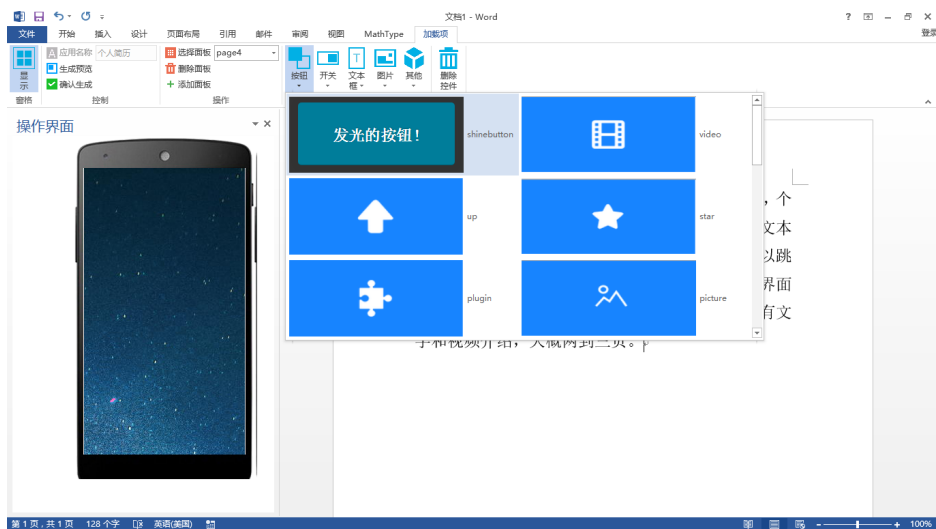


Fig. 12. 添加新控件

点击“确认生成”，会生成一个二维码，用手机扫描二维码即可下载安装生成的 APP



Fig. 13. 完成生成

5.2 使用 OfficeCoder 生成 App

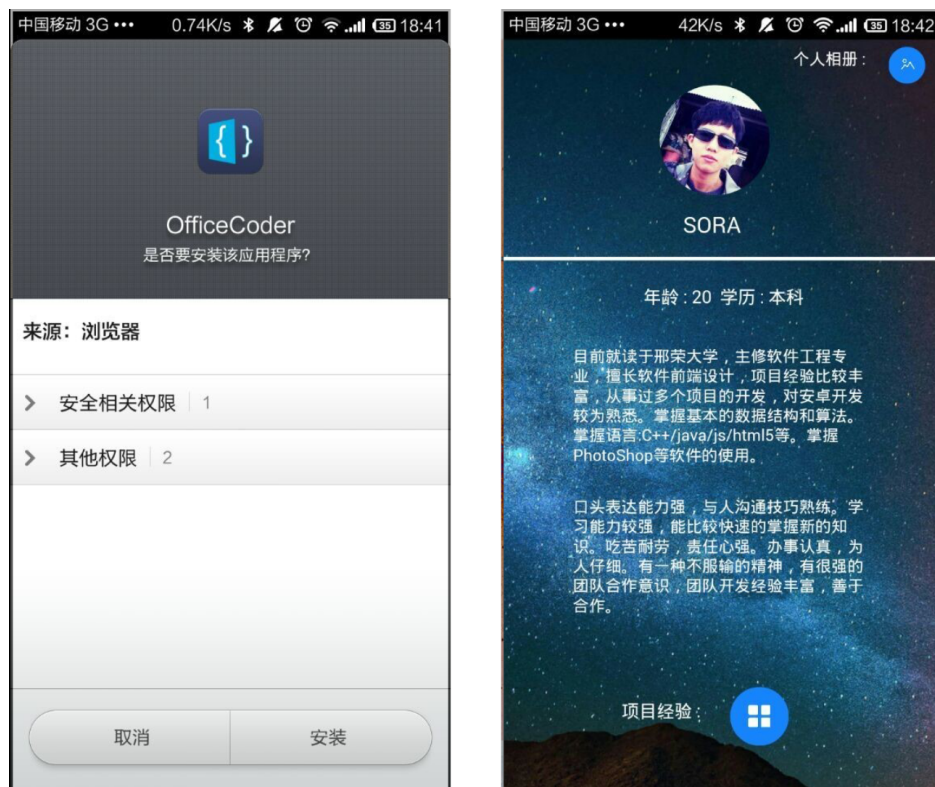


Fig. 14. 安装与使用

扫描 OfficeCoder 所生成的二维码, 安装下载好的 App, 打开安装的 APP 即可开始使用 OfficeCoder 所生成的 App

6 结论与展望

我们的 OfficeCoder 最终完成了可以通过编辑 Office 文档上传至云端服务器进行语义分析, 从而使得用户可以定制自己的一款移动 app 的云应用架构。我们搭建了一个完整的模板库和控件库使得使用 OfficeCoder 不仅可以生成应用, 还可以生成游戏, 同时我们将 OfficeCoder 打造成一个跨平台应用生成系统。我们的作品具有很大的突破性和创新性。在此之前, 只有程序开发人员才能编写 App, 而普通人只能使用别人开发的 App, 这样显然不能满足每一个用户的需求, 但是自己又没有编程的技术, 没有办法开发自己想要的 App, 相信大部分使用者都有过这样的想法。app 开发的技术要求将一大部分有软件开发热情的人

拒之门外，使没有技术的人望尘莫及。我们的作品打破了这一传统，拓展了 App 的开发途径，不再使用传统的编写代码的形式，而是采用自动生成与手动修改相结合的方式，首先为根据 Word 或者是 Powerpoint 甚至是 Visio 内容自动生成一款 App，用户可以在 office 界面中预览自动生成的 App，如果不满意 App 的内容，可以在 Office 中采用拖拽、删除、编辑等方式进行修改，直到自己满意为止。然后服务器将根据用户的编辑再次生成 App 供用户下载。这样就使得每一个使用 office 的用户都可以开发自己想要的 App 而且并不需要特别学习编程技术，这极大的降低了 App 制作的难度。

我们的应用作为 Word 2013, PowerPoint2013, Visio 2013 的一个插件，增补到应用的菜单栏中，使用插件时，可以根据 Word 当前内容直接生成 App；可以将 PowerPoint 中的图文资源发送至服务器，作为 App 生成的素材；可以通过 Visio 设计流程图对 App 的处理逻辑流程进行指导；或者用户对 App 的进行图形化的交互设定。同时我们建立了一个 App 应用的模板库，模板库建立在 Windows Azure 云平台之上，这些模板是由开发者开发完成，可以由第三方开发者开发后上传到云中，模板库中的 App 模板是系统生成 App 使用的原型，这些模板可以进行个性化定制。最终生成的 App 确定后，服务器完成代码的组装以及配置文件的生成，进行编译，编译完成后生成 App 安装包，提供给用户下载安装。

7 鸣谢

本课题在选题及研究过程中得到戴鸿君老师的悉心指导。戴老师多次询问研究进程，并为我们指点迷津，帮助我们开拓研究思路，精心点拨、热忱鼓励。戴老师一丝不苟的作风，严谨求实的态度，踏踏实实的精神，不仅授我们以文，而且教我们做人，虽历时一年，却给以终生受益无穷之道。对戴老师的感激之情是无法用言语表达的。

References

- [1] 张春霞，郝天永 - 系统仿真学报: 汉语自动分词的研究现状与困难 2005
- [2] 林颖，史晓东，郭锋 - 中文信息学报: 一种基于概率上下文无关文法的汉语句法分析 (2006)
- [3] 慕春棣，叶俊 - 软件学报: 用于数据挖掘的贝叶斯网络 (2000)
- [4] 方超，杨怡玲，黄宇 - 计算机应用: 基于谓词检测方法的上下文感知案例研究 (2013)
- [5] 宗成庆: 统计自然语言处理. 北京: 清华大学出版社, 2008: 103-146
- [6] 莫建文，郑阳，首照宇，等: 改进的基于词典的中文分词方法. 计算机工程与设计, 2013, 34(5): 1802-1807
- [7] Christopher D. Manning Hinrich Schutze: Foundations of Statistical Natural Language Processing[M]. The MIT Press Cambridge ,Massachusetts London ,England ,1999.
- [8] 刘迁，贾惠波: 中文信息处理中自动分词技术的研究与展望 - 计算机工程与应用, 2006 - 211.144.32.150
- [9] 岑咏华，韩哲，季培培: 基于隐马尔科夫模型的中文术语识别研究 - 现代图书情报技术, 2008
- [10] 周强，俞士汶: 一种切词和词性标注相融合的汉语语料库多级加工方法 - 《计算机研究与运用》，北京语言学院出版社