文章编号:2095-6134(2015)02-0252-07

航班预定口语对话系统的设计与实现*

陈振锋[1,2],杨晓昊[3],吴蔚澜[1,2],刘 加[3],夏善红[2†]

(1 中国科学院电子学研究所 传感技术国家重点实验室,北京 100190; 2 中国科学院大学,北京 100190;
3 清华大学电子工程系 清华信息科学与技术国家实验室,北京 100084)

(2014 年 3 月 24 日收稿; 2014 年 5 月 16 日收修改稿)

Chen Z F, Yang X H, Wu W L, et al. Design and implementation of mandarin spoken dialogue system for flight reservation [J]. Journal of University of Chinese Academy of Sciences, 2015,32(2):252-258.

摘 要 介绍一个航班预定口语对话系统的设计与实现,该系统允许用户通过普通话进行航班信息查询与预定.重点介绍口语对话系统中的口语语言理解.为了克服语音识别引入的识别错误导致语义理解错误的问题,提出基于词混淆网络的两阶段中文口语语言理解方法:首先从词混淆网络中选择 N 元文法作为分类特征,进行主题分类,并通过语义分类模型解析获取对应的语义树结构;然后利用基于规则的语义槽填充器抽取相应的语义槽属性-值.该方法是数据驱动的,训练数据的标记比较容易.实验在汉语航班预定领域进行,结果表明,在语音识别字错误率很高的情况下,该方法比传统的基于语法规则的语言理解方法更加鲁棒,在语义理解正确率方面有明显改善.

关键词 口语对话系统; 口语语言理解; 语义理解; 词混淆网络; 对话管理

中图分类号: TN912.3 **文献标志码:** A **doi:** 10.7523/j.issn.2095-6134.2015.02.015

Design and implementation of mandarin spoken dialogue system for flight reservation

CHEN Zhenfeng[1,2], YANG Xiaohao[3], WU Weilan[1,2], LIU Jia[3], XIA Shanhong[2]

(1 *State Key Laboratory on Transducing Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing* 100190, *China*; 2 *University of Chinese Academy of Sciences, Beijing* 100190, *China*; 3 *Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing* 100084, *China*)

Abstract We present a spoken dialogue system for flight reservation, which allows users to inquire information about flight in mandarin. We describe the design and the implementation of our system, focusing on spoken language understanding (SLU). Considering that the speech recognizer inevitably makes errors, we propose a new two-stage mandarin SLU approach based on word confusion network. Firstly, the semantic tuple classifier is used to identify the topic of an input utterance using N-gram features extracted from the word confusion network and to parse a semantic tree by recursively calling semantic classification models. Then the rule-based semantic slot filler is

* 国家自然科学基金(61005019,61273268,90920302)和北京市自然科学基金(KZ201110005005)资助

† 通信作者,E-mail: shxia@mail.ie.ac.cn

used to extract the corresponding slot/value pairs. The advantage of the proposed approach is that it is mainly data-driven and requires minimally annotated corpus for training. Experiment has been carried out in the Chinese flight reservation domain, which shows that the proposed approach is more robust to speech recognition errors than the conventional handcrafted rule-based parser, and substantially improves performance of accuracy when the ASR word error rate is high.

Key words　spoken dialogue system; spoken language understanding; semantic analysis; word confusion network; dialogue management

口语对话系统(spoken dialogue system，SDS)是指通过语音实时地与人类进行智能对话的系统. 口语对话系统是语音识别、语言理解、对话管理、自然语言生成以及语音合成技术的集大成者，是语音识别技术走向实用阶段的一个重要研究领域. 口语对话系统使得用户可以在获取信息的同时，留出双手做更加重要的事情，一个明显的例子就是车载口语对话系统[1]，驾驶员需要用双手去控制车子的正常行驶，通过口语对话系统进行车载导航、通信等可以减轻驾驶员的负担. 目前，口语对话系统已经开始广泛应用到多个领域，如餐馆预定[2]、旅游信息查询[3-4]、天气预报信息查询[5-6]等.

随着电子设备日趋小型化，在任何时间、任何地点以更加快捷简便的方式获取信息变得越来越重要. 近年来，随着语音识别技术以及移动互联网的快速发展，语音技术相关的应用开始在智能手机上出现. 2011 年苹果公司推出 Siri 之后，市场轰动. 借助手机语音助手，用户可以通过语音命令进行搜索，远端的云计算服务器根据语音命令从网上数据库中提取相关的信息，并且将查到的信息变成语音、图像、文字等传送回来，使得用户可以非常轻松方便地获取有用信息.

在国内，百度语音助手、科大讯飞语音助手、搜狗语音助手也顺势推出，使得用语音打电话、发短信、设置闹钟，或者是查询天气、航班、出行路线信息变得可能. 图 1 展示了百度语音助手的一个使用示例. 当然目前的这些语音助手还不够智能，它们还只是用于特定对话领域，并且也达不到像人类间的交流那样自然. 但口语对话系统的广阔发展前景已经显现，随着研究机构和公司继续投入到口语对话系统的研究中，相信人们梦寐已求的人机无缝交流必将实现.

本文实现的航班预定口语对话系统，主要目的是帮助用户以口语对话的方式进行航班预定，

图 1　百度语音助手使用示例

Fig. 1　Usage example of Baidu voice assistant

用户可以输入出发城市、到达城市以及出发日期进行航班信息查询，并可以就航班所属的航空公司、票价、航班号、出发时间等进行询问，最终完成航班的预定.

本文介绍的航班预定口语对话系统主要侧重于口语语言理解、对话管理等方面的研究，主要创新工作包括: 1) 提出了基于词混淆网络的两阶段中文语义理解方法; 2) 实现了基于任务和基于填表的混合对话管理方法.

1　系统结构

图 2 是典型的航班预定口语对话系统的框图，该系统主要包含 5 个模块: 语音识别器、语义解码器、对话管理器、自然语言生成器和语音合成器. 其中，语音识别的主要任务是将用户语音转化成文字形式; 一旦获取了文字表达，语义解码器将基于文字进行语义分析，获取用户的意图以及系统感兴趣的信息交给对话管理器进行后续处理; 对话管理模块结合上下文对话历史，按照一定的策略组织对话流程，决定系统的响应，并从后台服务器中查询用户所需的航班信息，产生系统应答;

自然语言生成器将系统应答转化成自然语言，最后由语音合成器生成语音反馈给用户.

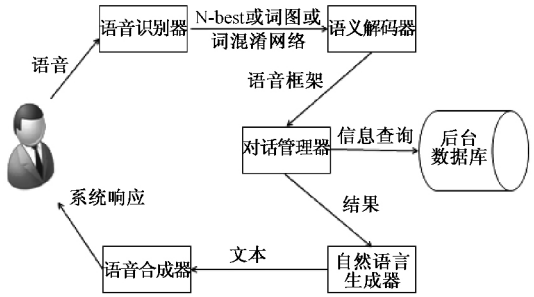

图 2 对话系统框图

Fig. 2 Block diagram of spoken dialogue system

1.1 词混淆网络

语音识别(automatic speech recognition，ASR)的输出结果主要有最优候选识别结果(1 - best)、多候选识别结果(N-best)、词图(word lattice)以及词混淆网络(word confusion network). 目前，大多数口语语言理解算法是对语音识别输出的最优候选进行语义解码，不过由于语音识别可能出现识别错误，使得语义理解性能退化. 对于电话语音，语音识别字错误率(word error rate，WER)大约在 34%，因此实用的语言理解算法应该具有好的鲁棒性，能够尽量容忍语音识别错误带来的干扰.

为了提高语言理解的性能，基于语音识别最优候选之外的新的算法被提出. 例如，CMU 的 Phoenix[7-8] 支持对 N-best 进行语义理解，Oerder 和 Ney[9] 提出将词图作为语音识别和语义理解的桥梁，Tür 等[10] 提出使用词混淆网络的语义理解方法.

词混淆网络是 1999 年由 Mangu 等[11] 首次提出，词混淆网络是一种归一化的词图，词图和词混淆网络的结构如图 3 所示. 词混淆网络将词图中互相混淆的词划分为同一组，对词图中同一时间点附近的词进行强制对齐，词混淆网络中的每个词都带有后验概率，可以当作该词的置信度分数，这一点对于口语对话系统来说非常有用. 语义理解模块从词混淆网络中抽取的语义概念包含有置信度分数，在置信度分数较低时，可以选择合适阈值将语义理解结果进行丢弃，也可以选择让对话管理模块向用户进行确认，从而提升对话系统的整体性能.

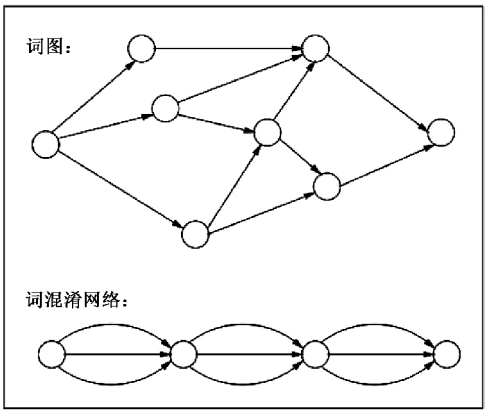

图 3 词图和词混淆网络的典型结构

Fig. 3 Typical structures of word lattice and word confusion network

文献[12]对语义理解算法使用最优候选、词图、词混淆网络进行性能比较，实验结果表明在主题分类(topic classification)以及语义槽提取(paramater extraction)中使用词混淆网络的性能最好，因此本文也选择词混淆网络作为语音识别和语言理解的接口.

1.2 基于词混淆网络的两阶段中文语义理解方法

语义理解是口语对话系统的关键技术之一，它的任务是理解用户意图并抽取用户输入所包含的关键信息. 通常用户输入句子的语义可以用语义框架(semantic frame)表示. 一个语义框架通常包括：1)语义框架类型(frame type)，表示用户输入语句的主题；2)相应的一些语义槽(slot)，表示用户输入语句中的关键语义概念. 例如，如果用户说“我想买张去上海的机票”，则对应的意图就是“flight(destination = “上海”)”.

本文提出的中文语义理解框架分为 2 个步骤：首先，语义元组分类器从用户输入的词混淆网络中提取分类特征，并解析得到语义树；然后，通过基于规则的方法提取出相应的语义槽属性 - 值对.

1.2.1 语义元组分类器

语义元组分类器(semantic tuple classifier，STC)由 Mairesse 等[13] 在 2009 年提出，STC 基于输入串中的 N-gram 计数来检测对话主题以及语义槽属性 - 值对. STC 解码器采用支持向量机(support vector machine，SVM)训练一组分类器，其中：多类分类器(multi-class classifier)用来预测

主题,二值分类器(binary classifier) 用来预测某个槽属性 - 值对是否存在.

STC 算法将语义树划分成概念元组,概念元组由 n 个语义概念节点链接而成. 例如,语义树 flight(destination(city)) 包含 2 个长度为 2 的元组(即 flight →destination,destination →city) ,以及 1 个长度为 3 的元组(flight →destination →city) .

STC 训练过程需要对输入进行预处理,将输入语句中的某些子串替换为相应的语义类标记,语义类代表对话领域中的关键概念,如地名、日期等.

对例句"我想从北京飞往上海"使用 STC 解码器得到语义树的过程如图 4 所示.

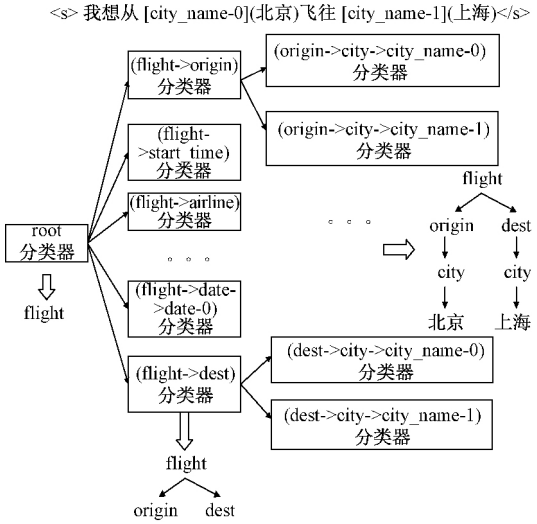

图 4　航班预定系统中对输入"我想从北京飞往上海"的语义树推导过程

Fig. 4　Semantic tree derivation for an utterance in the flight reservation system

STC 训练时,需要输入: 发音的语音识别结果 utt、对应的语义树标注以及领域数据库知识(定义语义类,预处理时使用) . 发音 utt 的特征用出现在发音中的 N 元文法 n-gram 表示,因此 utt 可以转化为特征向量 $\boldsymbol{X} = [x_1, \cdots, x_i, \cdots, x_{|\mathrm{NG}|}]$,第 i 个元素为

$$x_i = C_{\mathrm{utt}}(n\text{-gram}_i), \quad (1)$$

其中,$C_{\mathrm{utt}}(n\text{-gram}_i)$ 表示 N 元文法 n-gram$_i$ 在 utt 中出现的次数,n 取值为 1、2、3,即只考虑一元文法(unigram) 、二元文法(bigram) 和三元文法(trigram) ,而 | NG | 为应用领域的字表 n-gram 大小. 由于训练集中的 n-gram 只有很少一部分出现在输入 utt 中,因此特征向量 $\boldsymbol{X}$ 是稀疏的,这使得 SVM 的训练和分类过程都很快. STC 具体训练过程和分析算法见文献[13],本文不再赘述,SVM 训练采用线性核(linear kernel) ,使用 LibSVM 工具包[14] 进行训练.

1.2.2　语义槽属性 - 值抽取算法

由于口语对话系统需要语义槽的规整化值,因此用语义元组分类器对输入进行处理得到语义树后,还需要采用基于规则的方法抽取规整化的语义框架表示.

在航班预订口语对话系统中,语义概念主要涉及出发城市、到达城市、日期、时间、航班等,其中时间 - 日期表达方式非常多样,例如"明天上午","八月二号"等; 而且,有的时间 - 日期的具体意义需要由对话上下文决定,可能需要今天的具体信息才能判定其具体意义,例如,"星期五"的意义需要根据当天的日期才能确定是"本周五"还是"下周五",如果今天处于周五之前,那么"星期五"指的是"本周五",否则,更有可能指"下周五". 为此,本文参考 CMU 的 Phoenix[7] ,采用基于规则的方法对日期时间进行处理,然后通过递归算法将日期规整为 YYYY-MM-DD,时间规整为 HH: MM,目的是与后台航班信息数据库中的时间 - 日期格式保持一致. 图 5 给出了部分日期规则表示的例子.

```
[Date]
    ([Month] *[Day])   #几月几号
    ([Today_Relative])  #今天、明天、后天等
    ([Day_Name])        #星期一/二/.../日
;
[Month]
    ([Number]月)
;
[Day]
    ([Number]日)
;
```

图 5　日期规则表示的部分片段

Fig. 5　A portion of date grammar

例如,"8 月 2 日"符合下列语法:

$ date →$ month/MM 月/ < eps > $ day/DD 日/ < eps >

则可以将"8 月 2 日"抽取表示成"2014 - 08 - 02".

对于一些简单语义概念,如出发城市、到达城市等则可以直接从语义树中得到.

1.2.3　基于混淆网络的 STC 分类器

1.2.1 节介绍的 STC 分类器是采用最优候选

作为输入，当输入改成使用词混淆网络时，需要做一些调整. 式(1)中的 x_i 修改为

$$x_i = E(C_{\text{utt}}(n-\text{gram}_i))^{\frac{1}{|n-\text{gram}_i|}}, \quad (2)$$

式中，$|n\text{-gram}_i|$ 表示出现在 N 元文法 $n\text{-gram}_i$ 中词的个数，使用 $|n\text{-gram}_i|$ 的倒数作为指数进行归一化，因为越长的文法出现的可能性越小. $E(C_{\text{utt}}(n\text{-gram}_i))$ 是 $C_{\text{utt}}(n\text{-gram}_i)$ 的期望，由于词混淆网络中每个词都带有后验概率，因此计算出现次数时需要乘以每个词的后验概率，图 6 是词混淆网络的一个示例.

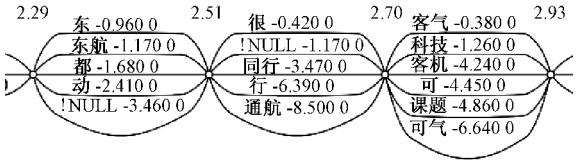

图 6 词混淆网络示例

Fig. 6 An example of word confusion network

1.3 基于任务和基于填表的混合对话管理方法

对话管理模块在口语对话系统中的主要任务是控制系统和用户之间的对话，主要包括：1) 根据语义理解模块对用户意图的理解以及对话上下文去后台数据库进行信息查询，并接收后台数据库查询结果；2) 根据任务要求，请求用户输入更多的槽信息以便提交给后台进行查询；3) 负责生成应答信息，并发送给自然语言生成子系统；4) 进行差错处理、主题切换，使得口语对话系统能按任务目标方向前进，并表现得更智能更像人[15].

目前，常用的对话管理方法有：1) 基于有限状态 finite state) 的方法[16]、2) 填表(form filling)的方法[17]、3) 信息状态更新(information state update)的方法[18-19]、4) 基于任务(task-based)的方法[20-21].

本文的航班预定系统中，用户在预定航班时主要需要提供出发城市、到达城市、出发日期这些必选项以便进行航班查询，然后再通过进一步的对话，比如航班号、出发时间、价钱等从而最终确定所选航班，因此考虑采用基于任务和填表的混合对话方式. 系统对话按照任务进行组织，如图 7 所示，其中，任务 GetInfo 采用填表方法，该任务用表(form)表示，表中包含一系列的槽，需要用户进行填写，每个槽对应一个系统提示(system prompt)用来向用户请求对应的信息，其中出发城市、到达城市、出发日期这些槽作为必填项，而航班号等作为可选填槽. 系统引导用户填满这些槽，在填槽过程中，用户也可以获取主动权，提供系统还没有问到的槽信息. 当所有必选槽都填满后，系统就可以组织后台数据库查询任务 GetResults，并根据填表结果执行 DiscussResults 任务，通过和用户交互，最终完成航班预定任务. 基于任务的对话管理方法有开源代码可供参考学习，因此本文只做简单思路介绍，具体实现可参考文献[22]，本文将填表方法和基于任务的方法相结合，目的是提高对话系统的自然度.

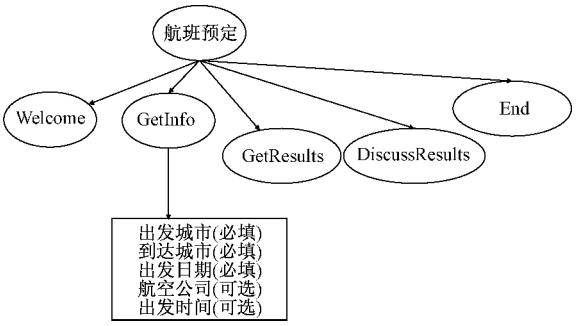

图 7 航班预定系统中对话任务

Fig. 7 Dialogue task in flight reservation system

2 实验

2.1 实验配置

本文实验中采用的语料为真实的国航订票数据，即用户通过电话与客服人员进行航班预定的语音标注，对话数据共有 26 242 句，真实数据中包含大量无关信息，通过人工筛选出有效数据进行语义标注，例如“从北京到上海的机票”标注为 flight(origin. city = “北京”，dest. city = “上海”)，“我跟您核对一下是三月六号星期三北京到长治然后六点五十起飞”标注为 flight(origin. city = "北京"，dest. city = “长治”，start_date = “三月六号星期三”，start_time = “六点五十”)，目前标注的句子共有 8 396 句，从中随机选取 6 000 条用于 SVM 训练，剩余的 2 396 句用于测试. 为了评估算法对语音识别错误的稳健性，数据集还包括对这些句子进行语音识别后的结果的测试，语音识别结果的词错误率为 34.5%.

2.2 实验结果

语义理解的评价指标主要包含 3 个：准确率 P，召回率 R，F - 度量，计算公式如下：

$$P = \frac{\text{正确的语义概念数}}{\text{提取的语义概念总数}},$$

$$R = \frac{\text{正确的语义概念数}}{\text{语料库中标注的语义概念总数}},$$

$$F = \frac{2 \times R \times P}{R + P}.$$

为了评估语言理解模块的准确性,对本文提出的基于词混淆网络的语义理解算法进行了测试,并将本文提出的方法和基于语法规则的 Phoenix 算法进行对比,实验结果如表 1 所示.

表 1　基于词混淆网络的两阶段语义理解方法和基于规则的 Phoenix 语义解码器性能比较

Table1　Performance comparison between the two-stage SLU method based on word confusion network and the handcrafted Phoenix parser　%

语义理解方法	准确率 P	召回率 R	F-度量
采用人工转写文本			
本文	98.1	94.8	96.42
Phoenix	97.9	93.5	95.65
采用语音识别结果			
本文	94.1	83.8	88.65
Phoenix	90.3	78.6	84.04

从表 1 可以发现,采用人工转写文本进行测试时,由于此时标注不是词混淆网络,因此本文提出的方法退化为原始的 STC 分类算法,此时特征向量用式 1 表示. 比较可以发现,本文提出的算法和基于规则的 Phoenix 语义理解结果性能差别不大; 在采用语音识别的结果进行测试时,由于语音识别的词错误率很高,与基于规则的 Phoenix 语义理解结果相比,本文算法性能有明显提高,F - 度量有大约 4.6% 的提示. 实验结果表明,本文提出的算法对语音识别错误有较好的容错性.

2.3　对话实例

目前我们实现了 2 套航班预定口语对话系统,一套是文本输入文本输出,作为初始演示版本; 另一套采用词混淆网络输入,这是因为目前实验室中使用的语音识别系统还是一套在服务器上通过脚本方式运行的系统,需要将录制好的 16 k 8 bit 的 wav 文件作为识别输入文件,生成词混淆网络. 用户通过麦克风输入语音、实时进行语音识别的口语对话系统将在以后的工作中完成,本文的主要工作集中在语义理解部分,演示示例中用户输入是先录好音,在服务器上识别得到词混淆网络后再作为对话系统的输入.

下面给出目前航班预定口语对话系统的一个实例,其中 S 表示系统,U 表示用户.

S: 您好,欢迎使用 FlightStar 航班预定系统

U: 我想买三月十三号从北京到上海的机票

S: 您好,2014 - 03 - 13 日,北京 - 上海的航班如下:

航空公司 起飞 - 降落时间　　起飞降落机场

吉祥航空 06:35 - 08:45　首都机场 T3　虹桥机场 T2

海南航空 07:50 - 09:55　首都机场 T1　虹桥机场 T2

东方航空 21:30 - 23:40　首都机场 T2　虹桥机场 T2

东方航空 22:00 - 23:55　首都机场 T2　虹桥机场 T2

您要买哪个航空公司的?

U: 买东航的航班

S: 您好,东方航空包含 2 趟,一趟 21:30 起飞,另一趟 22:00 起飞,请问想买那趟?

U: 晚上 9 点半的吧

S: 您好,请想购买的机票为:

出发地: 北京首都机场 T2

到达地: 上海虹桥机场 T2

航班: 东方航空

起飞时间: 21:30

降落时间: 23:40

3　结论

本文描述一个航班预定口语对话系统的设计与实现,重点介绍语义理解和对话管理模块. 口语对话系统中,语义理解是系统的核心,用于帮助系统正确理解用户意图,而对话管理模块也起着重要作用,它控制着整个系统和用户之间的交互过程,很大程度上影响用户对系统使用体验的感受,一个良好的对话管理模块还应该包含应答推理策略、差错处理策略等. 实验结果表明,本文提出的基于词混淆网络的两阶段语义理解方法具有很好的鲁棒性,在语音识别错误率很高的情况下,性能比基于规则的语义解码器性能有明显提高.

目前,我们实现的航班预定口语对话系统输入是基于语音识别的词混淆网络,输出是基于文

本的，下一步的工作重点是将可以直接与用户交互的语音识别以及语音合成整合到对话系统中.

参考文献

[1] Weng F, Cavedon L, Raghunathan B, et al. A conversational dialogue system for cognitively overloaded users [C] // International Conference on Spoken Language Processing. 2004.

[2] Liu J, Xu Y, Seneff S, et al. CityBrowser II: a multimodal restaurant guide in mandarin [C] // International Symposium on Chinese Spoken Language Processing. 2008: 1-4.

[3] Huang C, Xu P, Zhang X, et al. LODESTAR: a mandarin spoken dialogue system for travel information retrieval [C] // Proceedings of Eurospeech. 1999, 99: 1 159-1 162.

[4] 黄寅飞，郑方，燕鹏举，等. 校园导航系统 EasyNav 的设计与实现 [J]. 中文信息学报, 2001, 15(4): 35-40.

[5] Žibert J, Martinčić-Ipšić S, Hajdinjak M, et al. Development of a bilingual spoken dialog system for weather information retrieval [C] // Proceedings of Eurospeech. 2003: 1 917-1 920.

[6] Lin Y C, Chiang T H, Wang H M, et al. The design of a multi-domain mandarin Chinese spoken dialogue system [C] // International Conference on Spoken Language Processing. 1998.

[7] Ward W H. The Phoenix system: understanding spontaneous speech [C] // Proceedings of ICASSP. 1991, 66.

[8] Ward W, Issar S. Recent improvements in the CMU spoken language understanding system [C] // Proceedings of the workshop on Human Language Technology. Association for Computational Linguistics, 1994: 213-216.

[9] Oerder M, Ney H. Word graphs: an efficient interface between continuous-speech recognition and language understanding [C] // Proceedings of ICASSP. 1993, 2: 119-122.

[10] Tür G, Wright J H, Gorin A L, et al. Improving spoken language understanding using word confusion networks [C] // Interspeech. 2002.

[11] Mangu L, Brill E, Stolcke A. Finding consensus among words: lattice-based word error minimization [C] // Proceedings of Eurospeech. 1999.

[12] Hakkani-Tür D, Béchet F, Riccardi G, et al. Beyond ASR 1-best: using word confusion networks in spoken language understanding [J]. Computer Speech & Language, 2006, 20(4): 495-514.

[13] Mairesse F, Gasic M, Jurcícek F, et al. Spoken language understanding from unaligned data using discriminative classification models [C] // Processing of ICASSP. 2009: 4 749-4 752.

[14] Chang C C, Lin C J. LIBSVM: a library for support vector machines [J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27.

[15] Lee C, Jung S, Kim K, et al. Recent approaches to dialog management for spoken dialog systems [J]. JCSE, 2010, 4(1): 1-22

[16] Cole R. Tools for research and education in speech science [C] // Proceedings of the International Conference of Phonetic Sciences. 1999: 1 277-1 280.

[17] Aust H, Schroer O. An overview of the Philips dialog system [C] // DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, VA. 1998.

[18] Bos J, Klein E, Lemon O, et al. DIPPER: Description and formalisation of an information-state update dialogue system architecture [C] // 4th SIGdial Workshop on Discourse and Dialogue. 2003: 115-124.

[19] Ljunglöf P. trindikit. py: an open-source Python library for developing ISU-based dialogue systems [J]. Proceedings of IWSDS, 2009, 9.

[20] Rich C, Sidner C L. COLLAGEN: a collaboration manager for software interface agents [J]. User Modeling and User-Adapted Interaction, 1998, 8(3/4): 315-350.

[21] Bohus D, Rudnicky A I. The RavenClaw dialog management framework: architecture and systems [J]. Computer Speech & Language, 2009, 23(3): 332-361.