gAnswer 系统使用手册

目录

1 前言

知识库问答（knowledge base question answering, KB-QA）即给定自然语言问题，通过对问题进行语义理解和解析，进而利用知识库进行查询、推理得出答案。具体的，从应用领域的角度划分，知识库问答可以分为开放域的知识问答，如百科知识问答，和特定域的知识问答，如金融领域，医疗领域，宗教领域等，以客服机器人，教育/考试机器人或搜索引擎等形式服务于我们的日常生活。而 RDF 则是表示知识库的一种重要手段。

RDF（Resource Description Framework，资源描述框架）是由 W3C 提出的一组标记语言的技术规范，用来表现万维网上各类资源的信息并发展语义网络。在 RDF 模型中，每个网络对象都由一个唯一命名的资源来表示，用一个 URI（Uniform Resource Identifer ，统一资源标识符）来标识。 RDF 也利用 URI 去命名资源的属性和资源间的关系，以及关系的两端（通常被称为"三元组"）。因此，一个 RDF 数据集可以由一个有向、有标签的图来表示，其中资源是顶点，三元组是标签为属性或关系的边。

基于 RDF 表示的知识库数据，我们实现了基于海量知识库的自然语言问答系统，称为 gAnswer，这是北京大学的研究项目，并且由北京大学计算机科学与技术研究所的数据管理实验室对该系统进行开发和维护。若要了解 gAnswer 设计的详细描述，可以在**"出版物"**一章阅读我们的论文。这份帮助文档的其余部分包括系统的安装、使用、API、用例和 FAQ。gAnswer 目前已发布在 github 中，并遵循 BSD 协议。您可以使用 gAnswer、报告问题、提出建议，我们将与您一同使得 gAnswer 变得更好。您也可以在尊重我们的工作的前提下基于 gAnswer 开发各种应用。

请确保在使用 gAnswer 之前已经阅读了**"法律问题"**一章。

2 开始

2.1 快速导览

gAnswer 系统是一个基于海量知识库的自然语言问答系统，针对用户的自然语言问题，能够输出 SPARQL 格式的知识库查询表达式以及查询答案的结果。整个项目用 JAVA 编写，因此可以在不同操作系统上使用。

若您需要获取源码，可以从我们的 github 页面上获取：https://github.com/pkumod/gAnswer

2.1.1 开始使用

使用 jar 包部署

我们推荐您使用我们提供的打包好的 jar 文件部署 gAnswer，具体步骤为：

1. 下载 Ganswer.jar 与 data.rar 两个文件，我们推荐您从 github 的 release 页面下载最新版的 Ganswer.jar 与 data.rar，以保证稳定性。
2. 在控制台下解压 Ganswer.jar，您可以解压到任意文件路径下，但请保证 Ganswer.jar 文件与解压得到的文件处在统一路径下。
3. 在控制台下解压 data.rar，您需要把解压得到的文件夹置于 Ganswer.jar 文件所在的路径下。
   这时，您的文件结构应该如下所示：
   ./
   ++ addition
   ++ application
   ++ data
   ++ fgmt
   ++ jgsc
   ++ lcn
   ++ lib
   ++ log
   ++ META-INF
   ++ nlp
   ++ paradict
   ++ qa
   ++ rdf
   ++ utils
   ++ Ganswer.jar
4. 在控制台下运行 Ganswer.jar，等待系统初始化结束，出现 Server Ready！字样后，则说明初始化成功，您可以开始通过 Http 请求访问 gAnswer 的服务了。

使用 eclipse 运行

当您使用 eclipse 运行 gAnswer 系统时，只需要通过 clone 或者 download 获取工程源码，然后按正常步骤导入 Eclipse 工程，同时将 lib 中的 jar 包加入 Build Path 中即可。

运行

目前我们仅仅提供了 HTTP API，以 JSON 格式接受用户自然语言问题，同时以 JSON 格式返回生成的 SPARQL 查询和问题答案。具体请参考 2.5.2 **gAnswerHttp** 一章和 3.1 **HTTP API 说明**一章

2.2 系统要求

数据集。gAnswer 系统需要使用 RDF 格式的数据集，目前我们使用 dbpedia 2016 数据集作为 gAnswer 的知识库，我们对 dbpedia 2016 的官方数据进行了一定的筛选和预处理。

外部图数据库系统。gAnswer 系统的运行需要借助支持 SPARQL 查询的图数据库系统来获取最终答案。我们在目前的版本中，我们使用 gStore 系统，并且在其基础上利用预处理的 dbpedia 2016 数据进行建库，并使用 http 请求与其交互。关于 gStore 系统，详情请参阅其 github 页面 https://github.com/pkumod/gStore

外部工具包。gAnswer 系统在问题理解阶段需要借助一些 NLP 工具，主要包括 maltparser、StanfordNLP，在 sparql 生成阶段，需要借助 Lucene 对辅助信息进行索引。

其他。见下表：

项目	要求
操作系统	Linux，Windows
架构	x86_64
磁盘容量	>8GB
内存空间	>20GB
Java	版本 >= 1.6

2.3 基本介绍

2.3.1 什么是 gAnswer

随着网络上的结构化数据日益丰富，其中，以 RDF 数据为重要代表。如何充分利用这些结构化知识逐渐成为了一个重要的课题。尽管 SPARQL 可以很好地处理在 RDF 数据上的查询，但是，由于 SPARQL 语法与 RDF schema 的复杂性，未经训练的普通用户在使用 RDF 数据时往往遇到困难。因此，在 NLP 和数据库领域，基于 RDF 知识库的自然语言问答系统受到了关注。于是，我们开发了 gAnswer 系统。gAnswer 能够将自然语言问题转化成包含语义信息的查询图，然后，将查询图转化成标准的 SPARQL 查询，并将这些查询在图数据库中执行，最终得到用户的答案。值得一提的是，我们在生成查询图的阶段，保留了自然语言中的歧义信息，把歧义的解决放到答案生成的步骤中。总而言之，gAnswer 系统主要有以下 3 个特点：

1. gAnswer 结合了查询解析和歧义消除功能。
2. gAnswer 基于子图匹配来获取用户答案。
3. gAnswer 提出了一种基于图的数据挖掘方法，生成了谓词词典，作为系统的辅助信息。

2.3.2 开源与授权

gAnswer 的源代码遵循 BSD 开源协议。你可以使用 gAnswer 、报告建议或问题，或者加入我们使 gAnswer 变得更好。在尊重我们的工作的前提下，你也可以基于 gAnswer 开发各种应用。

2.4 安装指南

2.4.1 包依赖

在目前版本的 gAnswer 系统中，需要引入如下 jar 包:

commons-codec-1.3.jar
commons-httpclient.jar
commons-logging.jar
GstoreJavaAPI.jar
jetty-all-9.0.4.v20130625.jar
json.jar
liblinear-1.8.jar
libsvm.jar
log4j.jar
lucene-core-2.0.0.jar
maltparser-1.9.1.jar
mysql-connector-java-5.1.7-bin.jar
Stanford-corenlp-1.3.4-models.jar
Stanford-corenlp-1.3.4.jar
xom.jar

2.4.2 外部接口依赖

在目前的 gAnswer 系统中，我们需要借助一些外部系统接口。在我们公开的版本中，我们提供了远程的外部系统调用函数，因此，您并不需要在您的计算机上安装这些外部系统。当然，您也可以选择自己安装它们。具体见下表:

项目	要求	位置
gStore	版本 >= v0.7.0	qa.GAnswer.getAnswerFromGStore2()
DBpediaLookup		qa.mapping.DBpediaLookup

出于性能考虑，我们强烈建议您使用自己在本地安装的 gStore 和 DBpediaLookup 服务。

如果您需要安装 gStore，请从 gStore 的 github 主页上获取资源与相关信息:
https://github.com/pkumod/gStore

如果您需要安装 DBpediaLookup，请从 DBpedia 的官方网站上获取资源与相关信息:
https://wiki.dbpedia.org/lookup/

当您选择使用自己配置的 gStore 或 DBpediaLookup，您需要下载我们的源码，自行更改源码中 gStore 与 DBpediaLookup 的服务器地址与端口（需要修改的位置见上表），才能够正确使用 gAnswer。需要注意的是，这时您只能从 IDE（我们推荐您使用 Eclipse）运行 gAnswer 工程。如果您仍然需要使用 jar 包运行 gAnswer，您需要自行打包 jar 包。关于如何导出 jar 包，我们在附录 A 中给出了一个在 eclipse 下操作的教程。

2.4.3 文件依赖

在 gAnswer 系统的运行中，需要借助一些辅助信息，它们以文件形式存放在硬盘上，并在初始化过程中被加载进内存中。您可以直接从我们提供的链接下载这些外部文件，也可以自行生成这些文件。这些文件具体包括：

文件/文件夹	描述
16predicate_id	RDF 数据中出现的谓词到编号的映射
16entity_id	RDF 数据中出现的实体到编号的映射
16basic_types_id	RDF 数据中出现的 rdf type 到编号的映射
16yago_types_list	RDF 数据中出现的 yago type 的列表
16type_id_all	所有出现的 type 到编号的映射
16type_fragment	数据集中各种 type 所包含的实体集合
16entitiy_fragment	数据集中每个实体可以接受的实体和谓词
16predicate_fragment	数据集中谓词可以接受的 type 类型
16dbo_predicates	数据集中属于 dbo 域的谓词列表（标准谓词）
paraphrase_dictionary	谓词复述词典
stopEntDict	停用的实体名称列表
dbpedia-relation-paraphrases-withScore-baseform-merge-sorted-rerank-slct	一些常见的自然语言模式到 RDF 数据中标准的谓词的映射

2.5 如何使用

2.5.1 数据格式

目前，我们仅仅支持通过 gAnswerHttp API，使用 http 请求和 json 格式来与 gAnswer 进行交互。具体的数据传输格式，请参阅“3 高级”一章中的“3.1.2 API 结构”部分。

2.5.2 gAnswerHttp

gAnswerHttp 是利用 jetty 开发的，嵌入式的，轻量级的 gAnswer http server。在控制台/终端启动 gAnswerHttp 以后，可以通过 http 请求获取系统生成的问题 sparql 和问题答案。

启动的方法是：切换到工程文件夹下，控制台输入 java -jar ../GanswerHttp.jar，默认为 9999 端口，Dictionary 为加载的词典，默认为目前系统自带的词典。

3 高级

3.1 HTTP API 说明

3.1.1 简单样例

在本部分，会给出一个使用 gAnswerHttp API 的例子。

正常启动 gAnswerHttp 后，用户需要构建一个 json 格式的数据，例如：

{

“maxAnswerNum”: “3”

“needSparql”: “2”

“question”: “Who is the wife of Barack Obama?”

}

上述 json 数据的含义为：回答”Who is the wife of Barack Obama?”这个问题，要求最多返回 3 个不同的答案，1 条生成的 SPARQL 查询。

将此 json 数据转化为字符串，进行 url 转码，然后使用 ip:port/gSolve/?data=%json string%（在%json string%处放入 json 数据字符串）这一 uri 来调用 gAnswer 系统。在样例中，实际访问的 uri 为：

http://ip:port/gSolve/?data={maxAnswerNum:3,needSparql:1,questions:Who is the wife of Ming Yao?}

假如系统返回了正确结果，返回的内容也是 json 格式的数据，如下所示：

（注意，使用浏览器直接访问时，应在**源代码**页面查看返回结果，避免出现显示错误）

{

“status”: “200”

“query”: “Who is the wife of Barack Obama?”

“vars” : [“?wife”]

“sparql”: [

“select DISTINCT ?wife where { ?wife <spouse> <Yao_Ming>. } LIMIT 3”

]

“results”:{

“bindings” : [

{

“?wife” :

{

“type” : “uri”

“value”: ”<Ye_Li>”

}

},

]

},

}

需要特别说明的是，其中“vars”代表识别到的变量名，“results”中为实际得到的答案，”value”中为实际答案的值，“status”则说明这是一次正常的请求返回。

假如系统中出现了错误，那么同样会返回 json 格式的数据，而不会报错。如下：

```
{
    “query”: “”,
    “message”: “UnvalidQuestionException: the question you input is invalid, please check”,
    “status”: “500”
}
```

这是输入了无效问题（长度过短）以后，返回的 json 数据，其中“message”提供了出错的详细信息，“status”说明了错误的代码。

3.1.2 API 结构

gSolve

uri

ip:port/gSolve/

作用

向 gAnswer 请求返回一个或多个用户问题的答案和 sparql，用户可以选择返回答案还是 sparql

输入

```
{
    “maxAnswerNum”: //一个整数，代表需要返回的答案的上限
    “needSparql”: //一个整数，代表需要返回 sparql 的数量
    “questions”: //用户问题
}
```

输出

```
{
    “status”: //表示用户请求的状态
    “query”: //用户问题
    “vars” : [ …… ] //查询中涉及的变量
```

```
    “sparql”: [ …… ] //由用户问题得到的 sparql 查询
    “results”:{
     “bindings” : [ //多个 json 对象，每个对象为一组变量的值绑定
       {
            “%varName%” : { //此处的 key 为上面“vars”中出现的变量的名字
              “type” :  //说明这个值的类型，比如 uri
              “value”:  //答案的值
            }
       }
    ]
    },
    //当 maxAnswerNum 大于 0，会返回得到的答案
}
```

getInfo

uri

ip:port/getInfo/

作用

获取当前 gAnswer 系统的状态以及相关元数据

输入

不需要额外的参数

输出

```
{
    “version”: //当前的系统版本
    “dataset”: //当前使用的数据集
    “GDB system”: //当前使用的 gStore 版本
}
```

3.1.3 Java API 示例

我们在源代码的 application.gAnswerHttpConnector 中给出了使用 Java 通过 http 请求

访问 gAnswer 系统的示例。

3.2 出版物

[1]. Sen Hu, Lei Zou, Haixun Wang, Jeffrey Xu Yu, Wenqiang He: Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs. IEEE TKDE 2017 [pdf]

[2]. Shuo Han, Lei Zou, Jeffrey Xu Yu, Dongyan Zhao: Keyword Search on RDF Graphs - A Query Graph Assembly Approach. CIKM 2017: 227-236 [pdf]

[3]. Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, Dongyan Zhao: Natural language question answering over RDF: a graph data driven approach. SIGMOD Conference 2014: 313-324 [pdf]

[4]. Ruizhe Huang, Lei Zou: Natural language question answering over RDF data. SIGMOD Conference 2013: 1289-1290 (undergraudate student's poster) [pdf]

[5]. Weiguo Zheng, Lei Zou, Xiang Lian, Jeffrey Xu Yu, Shaoxu Song, Dongyan Zhao. How to Build Templates for RDF Question/Answering: An Uncertain Graph Similarity Join Approach SIGMOD Conference, 2015 (to appear). [pdf]

3.3 FAQ

4 其他

4.1 贡献者

4.1.1 人员

邹磊（北京大学）　　项目领导

4.1.2 学生

胡森（北京大学）　　博士研究生
林殷年（北京大学）　硕士研究生

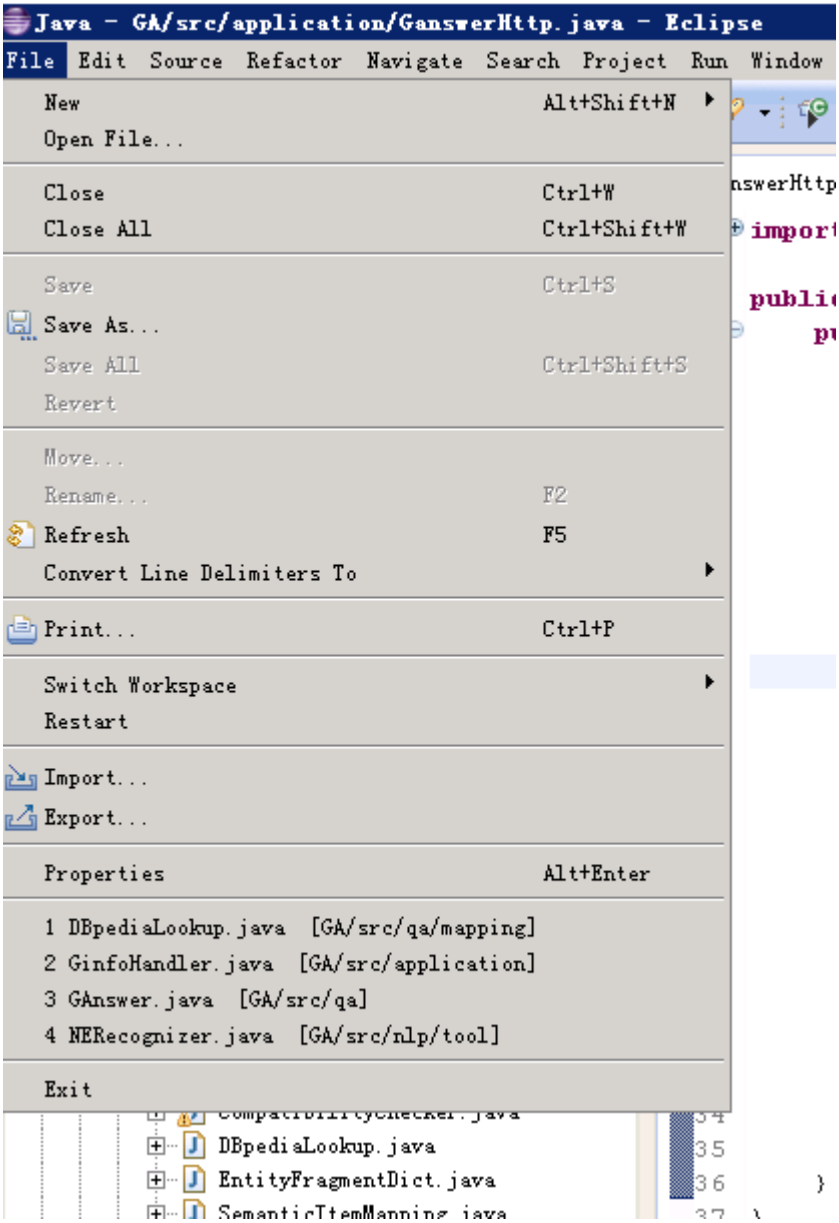
4.2 更新日志

v 0.1.0
系统上线

4.3 法律问题

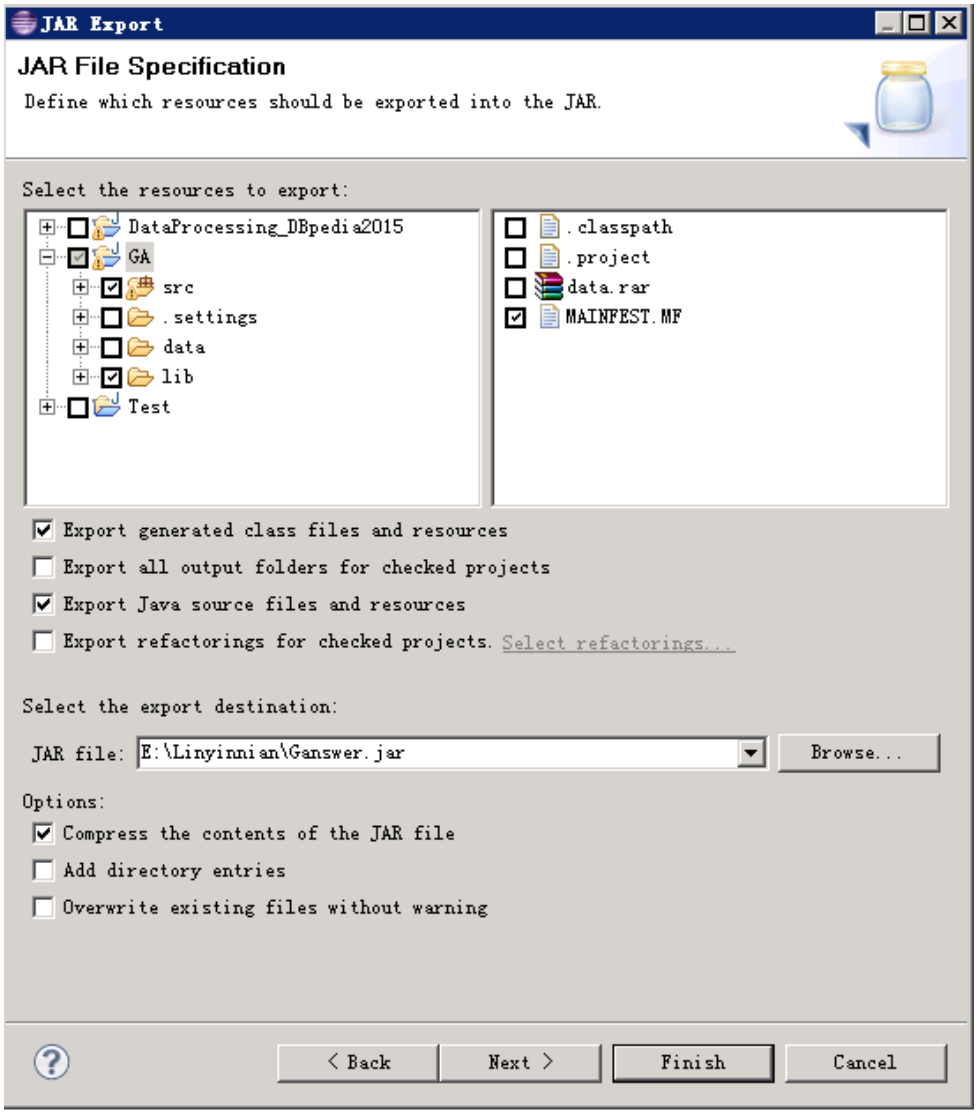
附录 A：在 Eclipse 中导出 jar 包

如果您需要在 Eclipse 中导出 jar 包，具体步骤为：
1. 在工程目录下，点击菜单中的 File->Export



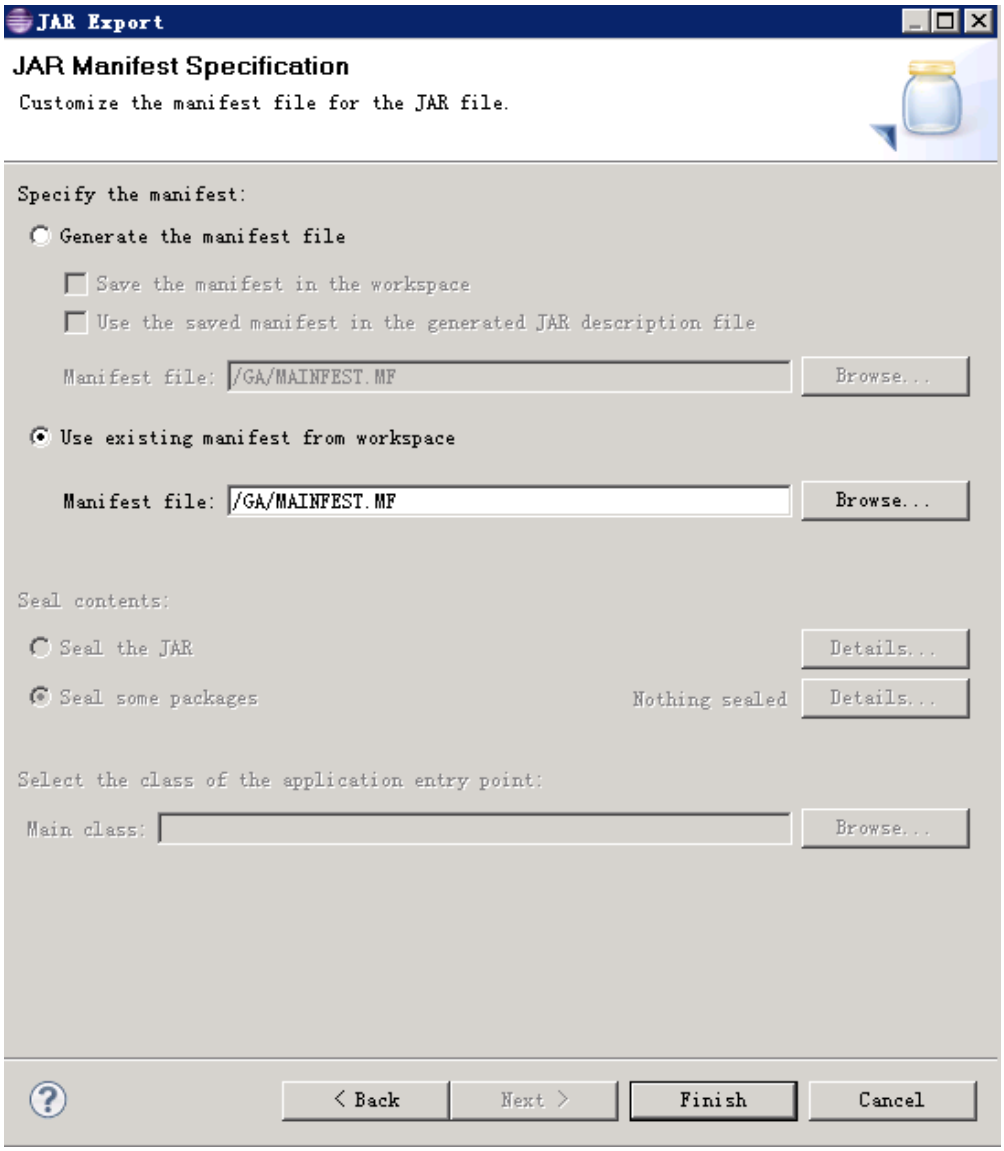
注意，这时您需要保证所有引用的外部 jar 包都处在 lib 文件夹中。

2. 在导出的文件类型中选择 JAR File，然后跳转到如下对话框，这时需要选择您要打包的工程文件内容，其中必须勾选的是 src 文件夹和 lib 文件夹，同时您可以给导出的 jar 包命名：



您可以注意到，在上图中 data 数据夹没有被勾选，因为 data 数据夹过大，可能会导致 jar 包生成失败。同时，您也可以选择不勾选 MAINFEST.MF 文件，只需要您在下一步中选择配置了该文件即可。

3. 连续点击两次 Next，来到 JAR Manifest Specification 页面，这是您需要勾选 Use existing manifest from workspace，然后选择工程文件夹下的 MAINFEST.MF 文件，这个文件规定了工程的主类和 classpath，如果您添加了新的主类或外部 jar 包，您需要修改这个文件。这个文件在格式上有特殊的要求，详情请参阅 Java 的官方帮助文档。



4. 点击 Finish，等待片刻后，就生成了需要的 jar 包，然后根据“从 jar 包部署”中给出的步骤部署您的 gAnswer 系统即可。