QuestionAnsweringSystem 技术实现

@mr.cc

QuestionAnsweringSystem[1]是一个 Java 实现的人机问答系统，能够自动分析问题并给出候选答案。本文从工作原理、主要数据结构、关键技术及代码实现四个方面对该系统的技术实现进行分析。

1、 工作原理

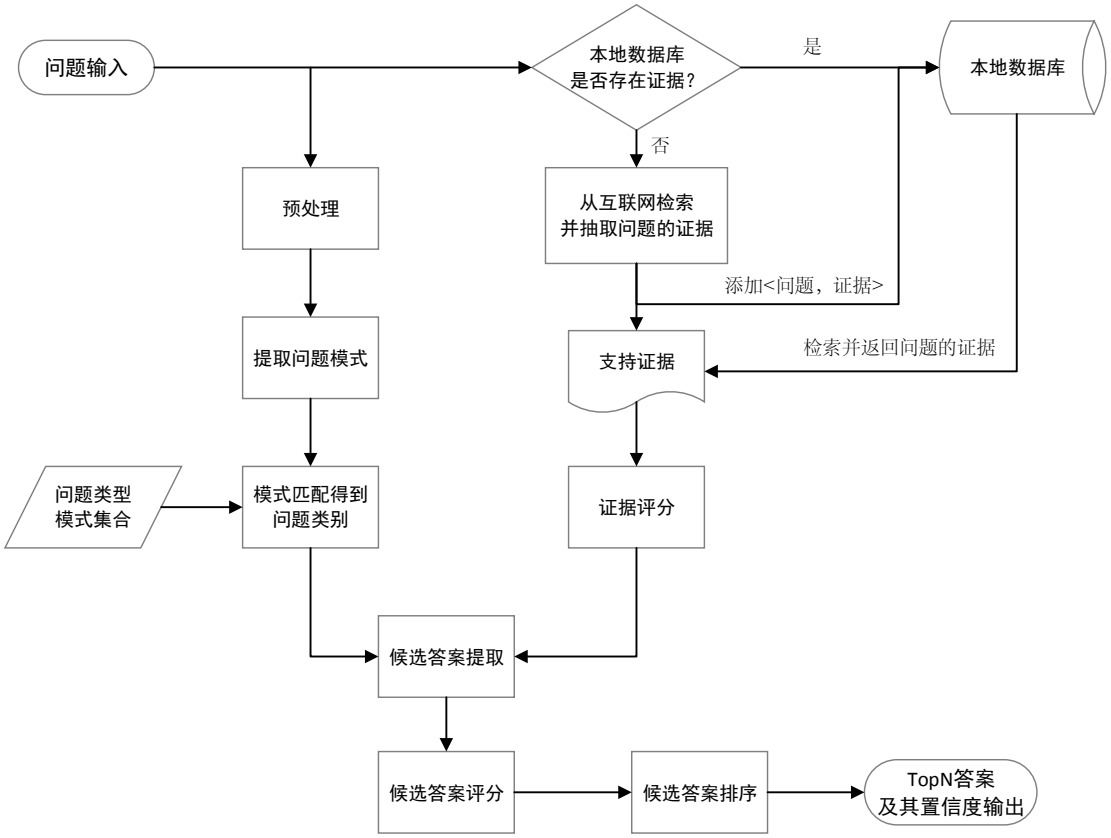


图 1 QA 系统工作原理

该 QA 系统的工作原理如图 1 所示，主要可以划分为证据获取、证据评分、问题分类、候选答案评分四大模块。这四大模块的目标及内容如下：

- ***证据获取***：从本地数据库或互联网上获取支撑问题的证据。若本地数据库存储有该问题的证据，则直接返回支撑问题的证据。否则，需要利用搜索引擎（如百度、谷歌）从互联网上抓取与该问题相关的片段，并抽取、整理出其中的正文作为该问题的支撑证据。
- ***证据评分***：为评价不同证据对问题的支撑度，需建立一套证据评分机制。

[1] 项目作者：杨尚川；项目地址：https://github.com/ysc/QuestionAnsweringSystem

证据评分模型，采用了基于词频的、基于 bigram 的和基于 skip-bigram 的三种评价方法及基于上述三种方法加权的组合方法。评分过程中，可以由用户设定以上四种评价方法的任意一种。

- ***问题分类***：对问题所属的类别进行判定。该系统将可识别的问题类别划分为人名、地名、机构名、数字、时间、定义和对象七类（暂时仅支持前五类），并预先定义这几类问题的匹配模式。分类过程为：1）提取问题的模式，2）和预定义的问题类型模式进行正则匹配，3）根据匹配的结果确定问题的类别。针对同一类问题，系统又定义了五大类数小类的匹配模式，用户可以自由设定匹配模式为大类别或基于五大类的加权组合。
- ***候选答案评分***：为评估候选答案的质量，需建立一套候选答案评分机制。在候选答案评分模型中，基础的评价方法有基于词频的、基于词距的、基于最短词距的、基于文本对齐的、基于宽松文本对齐的、基于回带文本对齐的和基于热词的七类方法；综合的评价方法有基于基础评价方法加权的组合方法。评价过程为：1）根据问题类型确定答案类型，然后从证据词集中筛选出命名实体标记与答案类型一致的词，作为候选答案，2）针对每个候选答案，利用评分模型进行打分，用户可以自由设定八类评价方法中的一种作为评分模型。在打分过程中，每类评价方法均有一个权值，候选答案的得分是评价方法的打分与该评价方法权值的乘积。

2、 主要数据结构

该 QA 系统最主要的数据结构有“问题（question）”和“证据（evidence）”两个，分别定义了输入的问题及支撑问题的证据的结构，描述如下：

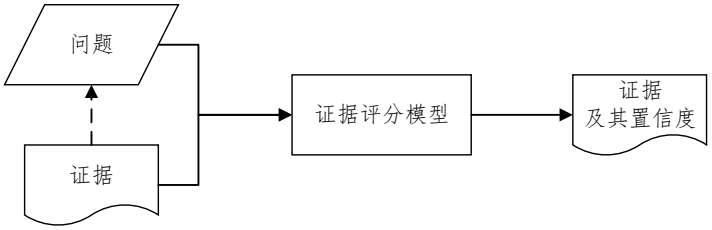
	属性	类型
question	问题	字符串
	支撑证据	evidence 列表
	问题类型	枚举
	预期答案	字符串
	候选问题类型	问题类型集合
	候选答案过滤器	方法
evidence	标题/title	字符串
	片段/snippet	字符串
	得分/score	Double
	候选答案	<answer, score>集合

3、 关键技术

该系统涉及的关键技术包括预处理（分词、词性标注及依存句法分析）、证据评分模型、候选答案评分模型[2]和问题分类模型。其中，预处理采用了开源工具(分词+词性标注：anjseg-0.9[3]，依存句法分析：stanford-parser-3.3.1[4])，本节将不会对预处理技术作特别说明。

3.1 证据评分模型

该模型的目标是评价支撑问题的证据的可信度，包含三个子模型（基于词频、基于 bigram、基于 skip-bigram）和一个组合模型（前三个子模型的线性加权）。



- ***基于词频的评价模型***
  a) 对问题及支撑证据进行分词、去停用词等处理；
  b) 统计证据（包括 title 和 snippet）中所有词的词频$\{tf_i\}$；
  c) 将问题中的词与证据中的词匹配，对于问题中的词，若在证据的 title 中出现，记 2/tf_i分；若在证据的 snippet 中出现 1/tf_i分；
  d) 对问题中所有的词的得分累加求和，并乘以该评价模型的权重，得到该证据的评分。
- ***基于 bigram 的评价模型***
  a) 分词并提取问题的二元词，问题经分词后表示为$S = \{w_1 w_2 \cdots w_n\}$，S 的二元词集为$\{w_1 w_2, w_2 w_3, \cdots, w_{n-1} w_n\}$；
  b) 统计二元词集的词在证据（包括 title 和 snippet）中出现的次数，出现一次记 2 分；
  c) 对二元词集的所有词的得分累加求和，并乘以该评价模型的权重，得到该证据的评分。
- ***基于 skip-bigram 的评价模型***
  a) 分词并提取问题的 skip-bigram，问题经分词后表示为$S = \{w_1 w_2 \cdots w_n\}$，S 的 skip-bigram 词集为$\{w_1 . w_3, w_2 . w_4, \cdots, w_{n-2} . w_n\}$，词集里的点号表示任意字符；

2 http://brenocon.com/watson_special_issue/08%20textual%20evidence%20gatherint.pdf，证据及候选答案评分模型主要以 IBM Watson 系统的“文本证据收集与分析”为参考。
3 https://github.com/NLPchina/ansj_seg
4 http://nlp.stanford.edu/software/lex-parser.shtml

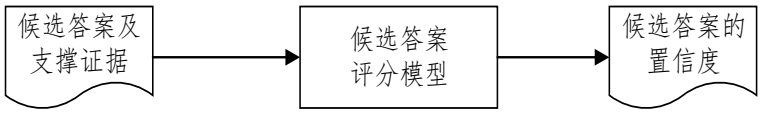
b) 统计 skip-bigram 词集的词在证据（包括 title 和 snippet）中出现的次数（正则匹配），出现一次记 2 分；

c) 对 skip-bigram 词集的所有词的得分累加求和，并乘以该评价模型的权重，得到该证据的评分。

在该 QA 系统中，各评价子模型的初始权重均设置为 1。

3.2 候选答案评分模型

该模型的目标是评价候选答案的可信度，包含七个子模型（基于词频、词距、最短词距、文本对齐、宽松文本对齐、回带文本对齐和热词）和一个组合模型（前七个子模型的线性加权）。候选答案，是支撑证据中命名属性与问题类型一致的词。每个评价子模型的初始权重均设置为 1。



- ***基于词频的评价模型***

a) 统计候选答案 c 在证据（包括 title 和 snippet）中出现的次数，若在 title 中出现，记两次；若在 snippet 中出现，记一次；

b) 累计候选答案在证据中得到的总次数，乘以该评价模型的权重，得到该候选答案的评分。

- ***基于（最短）词距的评价模型***

a) 给定候选答案 c；

b) 对问题 Q 和证据 E 进行分词处理，$\mathrm{Q}=\{t_1 t_2 \cdots t_m\}$，$\mathrm{E}=\{w_1 w_2 \cdots w_n\}$；

c) 计算 c 在 E 中的所有位置$\{c_i|\mathrm{i}=1,2,\cdots,\mathrm{u}\}$；

d) 分别计算 Q 中词项在 E 中的所有位置$\{t_{kj}|\mathrm{k}=1,2,\cdots,m; j=1,2,\cdots,\mathrm{v}_k\}$；

e) 候选答案与单个词项之间的距离$\mathrm{distance}_1(\mathrm{c},\mathrm{t}_k)=\sum_{i=1}^{i=u}\sum_{j=1}^{j=v_k}|c_i-t_{kj}|$，最短距离 $\mathrm{distance}_2(\mathrm{c},\mathrm{t}_k)=\min|c_i-t_{kj}|$；$\mathrm{i}=1,2,\cdots,\mathrm{u}$; $j=1,2,\cdots,\mathrm{v}_k$；

f) 候选答案的词距$\mathrm{dis}_1(\mathrm{c})=\sum_{k=1}^{k=m} distance_1(c,t_k)$，最短词距$\mathrm{dis}_2(\mathrm{c})=\sum_{k=1}^{k=m} distance_2(c,t_k)$；

g) 候选答案的评分 $score(\mathrm{c})=\mathrm{score}_{origin}(c)/dis(c)*weight$，其中$\mathrm{score}_{origin}(c)$是候选答案的原始得分，$weight$是当前模型的权重，$dis(c)$是候选答案的词距或最短词距。

- ***基于文本对齐的评价模型***

a) 给定候选答案 c；

b) 对问题 Q 进行分词处理，$\mathrm{Q}=\{t_1 t_2 \cdots t_m\}$；

c) 将 c 放到 Q 中的每一个位置，得到 m 个句子，也即 m 个严格的对齐模

式：$\{ct_1t_2\cdots t_m, t_1ct_2\cdots t_m, \cdots, t_1t_2\cdots ct_m\}$；

d) 在严格对齐模式的基础上，再建立 m 个模糊的对齐模式，构造规则如下：允许严格对齐模式中每个词项后接 0~5 个任意字符，如 $c.\{0,5\}t_1.\{0,5\}t_2.\{0,5\}\cdots t_m.\{0,5\}$；

e) 由 c)和 d)得到 2m 个对齐模式，将这些模式分别放到证据文本中进行正则匹配，并统计匹配到的句子数 count 及这些句子的总长 lens；

f) 候选答案评分 $score(\mathrm{c}) = \mathrm{questionLen/avgLen} * weight$，其中，匹配长度 $\mathrm{avgLen} = \mathrm{lens/count}$，questionLen是问题长度，$weight$是当前模型的权重。

- ***基于宽松文本对齐的评价模型***

该评价模型基本和基于文本对齐的评价模型一样，唯一一点不同的是该模型在对问题进行分词之后，忽略长度为 1 的词项。

- ***基于回带文本对齐的评价模型***

该模型相比基于文本对齐的评价模型，唯一不同的地方在于用于匹配的证据文本。该模型所用的证据文本，是通过 Google 搜索由问题和候选答案组成的关键词而得到的结果。

- ***基于热词的评价模型***

a) 对证据文本进行分词，并统计每个词项在证据文本中出现的次数；

b) 对问题进行分词，选出长度大于 1 的词项，并由 a)得到这些词项在证据文本中出现的次数，选择出现次数最高的词作为热词；

c) 找出离热词最近的候选答案，该候选答案的得分先翻倍，再乘以该评价模型的权重，并将其作为最佳候选答案。

3.3 问题分类模型

该 QA 系统的问题分类采用的是模式匹配的方法，其核心内容是建立起问题类型的匹配模式（以下简称“模式”）。这些模式可以分为三大类：直接匹配模式、基于问题分词的词与词性/词性的匹配模式和基于问题主谓宾的词与词性/词性的匹配模式。在建立模式之后，先按选定的某类模式提取问题的模式，再与所有问题类型模式正则匹配，最后将得票最多的问题类型作为问题的类别。下面用例句“APDlat 的发起人是谁？”对三类模式进行说明：

C1 直接匹配模式

示例	APDlat 的发起人是谁？
问题类型模式（部分）	Person 1990 年中国共产党的总书记是谁 Person 请问初唐四杰是哪四位 Location “海的女儿”是哪个城市的城徽 Location 阿尔及利亚首都是哪个城市 Organization 爱国华侨陈嘉庚出资兴建的大学是哪一所 Organization 电影界百花奖的主办单位是什么 Number 北京邮电大学的占地面积有多大 Number 北京大学占地多少平方米 Time 发现大庆油田在哪一年 Time 澳门回归祖国在哪一年

C2 基于问题分词的词与词性/词性的匹配模式

示例 (词+词性)	APDlat/en 的/uj 发起人/n 是/v 谁/RW.RWPersonSingle ？/w
示例 (词性)	en/uj/n/v/ RW.RWPersonSingle/w
问题类型模式（部分）	PersonOfDis->Multi1 .*(RW.RWOrdinaryMulti)/(nr|nr1|nr2|nrj|nrf).* PersonOfDis->Single1 .*(RW.RWOrdinarySingle)/(nr|nr1|nr2|nrj|nrf).* LocationOfDis->Multi1 .*(RW.RWOrdinaryMulti)/(ns|nsf).* OrganizationOfDis->Multi2 .*(RW.RWOrdinaryMulti)/(nt).* DefinitionDis->Comparison .*(N.Concept).*(RW.RWComparisonAdj).* DefinitionDis->DescriptiveMulti .*(V).*(RW.RWOrdinaryMulti).*(N.Concept).* DefinitionSOB->ConceptDescriptive1 .*(RW.RWDescriptive)/(V)/(N.Concept).* NumberAreaSOB->Area1 .*(V)/(RW.RWNumber)/(N.NumberAreaOfConcept).* NumberCodeSOB->Code1 .*(N.NumberCodeOfConcept)/(V)/(RW.RWNumber).* TimePeriodSOB->Age .*(RW.RWNumber)/(N.TimeOfConcept).* TimeDateSOB->Date1 .*(N.TimeOfConcept)/(RW.RWTimeDate).* ObjectSOB->Academic1 .*(RW.RWOrdinarySingle)/(N.ObjectAcademic).* ObjectSOB->Animal1 .*(V)/(RW.RWOrdinarySingle)/(N.AnimalOfConcept).* ObjectSOB->Art1 .*(RW.RWOrdinarySingle)/(N.ObjectArt).* ObjectSOB->Language1 .*(RW.RWLocationSingle)/(N.LanguageOfConcept).*

C3 基于问题主谓宾的词与词性/词性的匹配模式

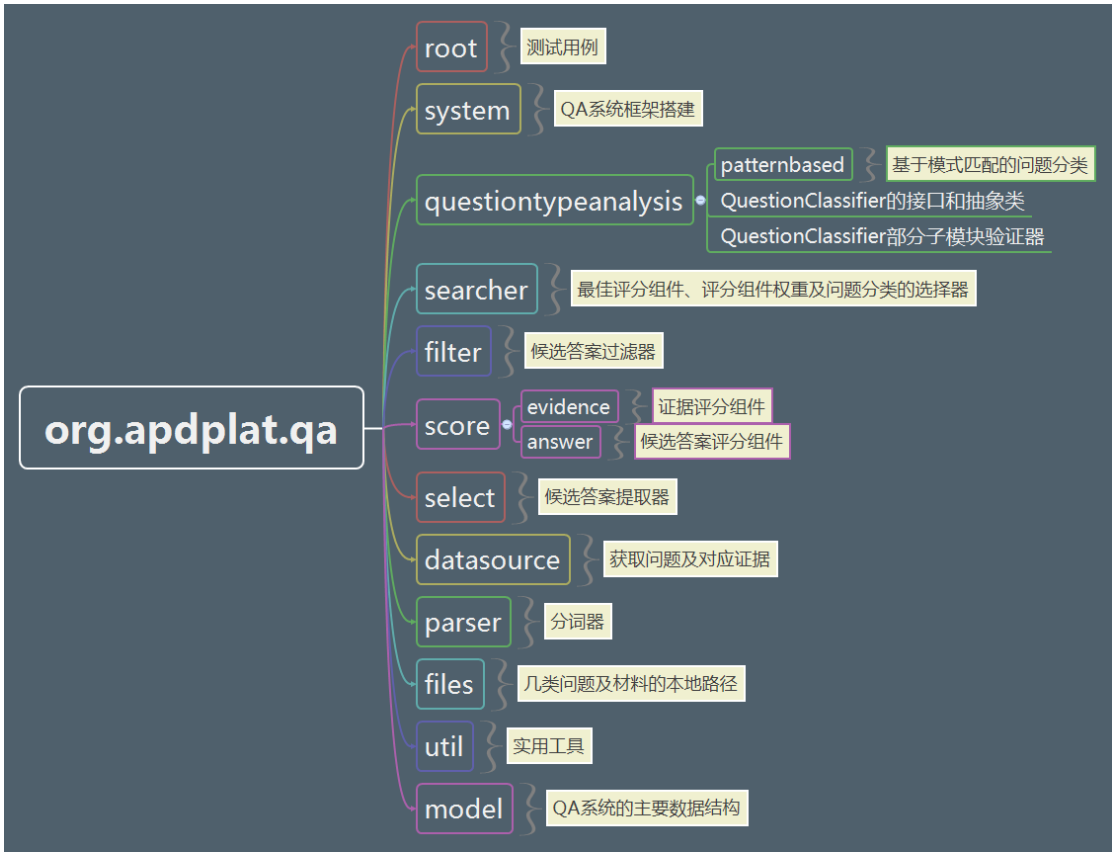
示例(词+词性)	发起人/n 是/v 谁/RW.RWPersonSingle
示例(词性)	n/v/ RW.RWPersonSingle
问题类型模式（部分）	Person->Multi2 .*(V.).*(RW.RWOrdinaryMulti).*(nr|nr1|nr2|nrj|nrf).* Person->Single1 .*(RW.RWPersonSingle).* Location->Multi2 .*(V.).*(RW.RWOrdinaryMulti).*(ns|nsf).* Location->Single2 .*(V.).*(RW.RWOrdinarySingle).*(ns|nsf).* Organization->Multi1 .*(V.).*(RW.RWOrdinaryMulti).*(nt).* Organization->Single1 .*(V.).*(RW.RWOrdinarySingle).*(nt).* Definition->Definition2 .*(N.Concept).*(V.).*(RW.RWDescriptive).* Definition->Language1 .*(N.LanguageOfConcept).*(V.).*(RW.RWOrdinarySingle).* Definition->Animal1 .*(N.AnimalOfConcept).*(RW.RWDescriptive).*(V).* Definition->Method1 .*(N.MethodOfConcept).*(V.).*(RW.RWOrdinarySingle).* Number->Ordinary1 .*(N.NumberNoun).*(RW.RWNumber).* Number->Area1 .*(N.NumberAreaOfConcept).*(RW.RWNumber).* Number->Code1 .*(N.NumberCodeOfConcept).*(V).*(RW.RWNumber).* Number->Frequency1 .*(RW.RWNumber).*(N.NumberFrequencyOfConcept).* Number->Temperature1 .*(ns|nsf).*(RW.RWTemp).* Time->Ordinary1 .*(V).*(RW.RWOrdinarySingle).*(N.TimeOfConcept).* Time->Date1 .*(N.Concept).*(RW.RWTimeDate).* Object->AcademicSingle .*(RW.RWOrdinarySingle).*(N.ObjectAcademic).* Object->AcademicMulti .*(RW.RWOrdinaryMulti).*(N.ObjectAcademic).* Object->ArtSingle1 .*(N.ObjectArt).*(V).*(RW.RWOrdinarySingle).* Object->Award .*(RW.RWOrdinaryMulti).*(N.ObjectAward).* Object->Category1 .*(RW.RWOrdinarySingle).*(N.ObjectCategory).*(V).* Object->Event1 .*(V).*(RW.RWOrdinarySingle).*(N.ObjectEvent).* Object->FoodMulti .*(N.ObjectFood).*(V).*(RW.RWOrdinaryMulti).* Object->Material1 .*(N.ObjectMaterial).*(V).*(RW.RWOrdinarySingle).* Object->TermMulti1 .*(N.ObjectTerm).*(V).*(RW.RWOrdinaryMulti).*

4、 代码实现

该系统是基于 Java 语言和 maven 环境搭建的，本节将从整体结构和主要类功能来说明其代码实现。

4.1 整体结构

该 QA 系统的代码实现整体结构如下图所示：



4.2 主要类功能

所在包	类名	功能
org.apdplat.qa.model	Question	问题的结构
	Evidence	证据的结构
	QuestionType	问题的类型，枚举
	CandidateAnswer	候选答案的结构
	CandidateAnswer-Collection	候选答案的集合
org.apdplat.qa.util	MySQLUtils	面向问题和答案的数据库操作
	TextExtract	基于规则方法的网页正文抽取
	Tools	混合工具包（比较杂）
	ZipUtils	Zip 压缩与解压缩

所在包	类名	功能
org.apdplat.qa.files	FilesConfig	几类问题及对应材料的本地路径
org.apdplat.qa.parser	WordParser	分词器
org.apdplat.qa.datasource	DataSource	获取问题及对应证据，接口，以下四个类均是该接口的实现类
	FileDataSource	从文件中获取问题及对应证据
	BaiduDataSource	利用百度搜索引擎从互联网上获取问题的对应证据
	GoogleDataSource	利用 Google 搜索引擎从互联网上获取问题的对应证据
	ConsoleDataSource	从控制台获取问题
org.apdplat.qa.select	CandidateAnswer-Select	候选答案提取组件，接口，下一个类是该接口的实现类
	CommonCandidate-AnswerSelect	通用候选答案提取组件
org.apdplat.qa.filter	CandidateAnswerFilter	候选答案过滤组件，接口，下一个类是该接口的实现类
	CandidateAnswer-CanNotInQustionFilter	候选答案过滤器(如果候选答案出现在问题中，则过滤)
org.apdplat.qa.score.evidence	EvidenceScore	证据评分模型，接口，以下四个类均是该接口的实现类
	TermMatch-EvidenceScore	基于词频的证据评分
	BigramEvidenceScore	基于 bigram 的证据评分
	SkipBigram-EvidenceScore	基于 skip-bigram 的证据评分
	Combination-EvidenceScore	组合证据评分
org.apdplat.qa.searcher	BestClassifierSearcher	根据已标注语料选择最佳分类器
	BestScoreSearcher	寻找最佳的证据和候选答案评分模型
	BestScoreWeight-Searcher	寻找最佳的证据和候选答案评分模型的权重

所在包	类名	功能
org.apdplat.qa.score.answer	CandidateAnswerScore	候选答案评分模型，接口，以下八个类均是该接口的实现类
	TermFrequency-CandidateAnswerScore	基于词频的候选答案评分
	TermDistance-CandidateAnswerScore	基于词距的候选答案评分
	TermDistanceMini-CandidateAnswerScore	基于最短词距的候选答案评分
	TextualAlignment-CandidateAnswerScore	基于文本对齐的候选答案评分
	MoreTextualAlignment-CandidateAnswerScore	基于宽松文本对齐的候选答案评分
	RewindTextualAlignment-CandidateAnswerScore	基于回带文本对齐的候选答案评分
	HotCandidate-AnswerScore	基于热词的候选答案评分
	CombinationCandidate-AnswerScore	组合候选答案评分
org.apdplat.qa.question-typeanalysis	QuestionClassifier	问题分类器，接口
	Abstract-QuestionClassifier	问题分类器，抽象类，实现上一接口
	QuestionType-Transformer	问题类型转换器，用于将类型模式文件中的问题类型转换为命名属性标记
	ValidateClassifier	依据已标注语料来验证基于模式匹配分类器的准确性
	ValidateMainPart-Extracter	依据已标注语料来验证主谓宾提取的准确性

所在包	类名	功能
org.apdplat.qa.question-typeanalysis.patternbased	PatternBasedMultiLevel-QuestionClassifier	继承了 AbstractQuestionClassifier，基于模式匹配的方法判断问题的类型
	MainPartExtracter	提取主谓宾
	QuestionPattern	问题的模式，枚举
	QuestionTypePattern-File	问题类型模式文件
	QuestionStructure	问题结构，用于提取主谓宾时的问题表示
	PatternMatchResultItem	模式匹配结果项的表示
	PatternMatchResult	模式匹配结果，包含模式的类别和匹配项
	PatternMatchResult-Selector	模式匹配结果选择器，接口
	DefaultPatternMatch-ResultSelector	默认的模式匹配结果选择器，是对上一接口的实现
	PatternMatchStrategy	模式匹配策略表示
org.apdplat.qa.system	QuestionAnswering-System	问答系统构造器，接口
	QuestionAnswering-SystemImpl	问答系统构造器的实现，是对上一接口的实现
	CommonQuestion-AnsweringSystem	通用问答系统构造器，继承了上一个类
	ScoreWeight	证据/候选答案评分组件的权重管理