

学 士 学 位 论 文

论文题目： 基于语义的成语检索方法研究

姓　　名： 雷舒婷

学 　号： 1231116

院 系： 信息科学学院

专 　业： 计算机科学与技术

指导教师： 荀恩东

二〇一六 年 六 月

**北京语言大学学士学位论文**

(2012级)

论 文 题 目： 基于语义的成语检索方法研究

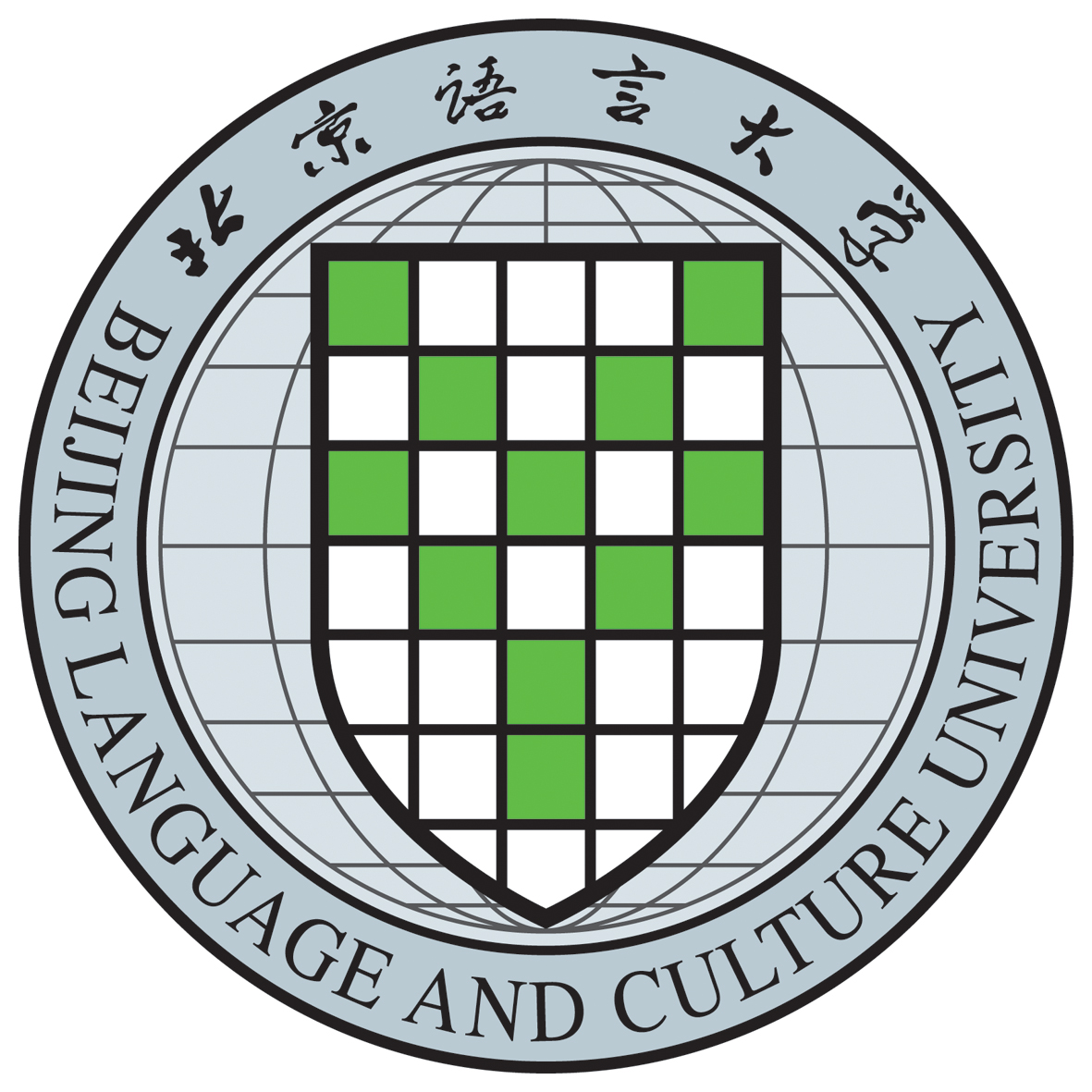
院 系： 信息科学学院

专 业： 计算机科学与技术

学 生 姓 名： 雷舒婷

指导教师姓名： 荀恩东

论文完成日期： 2016年6月



论文原创性声明

本人郑重声明：所呈交的论文，是本人在导师指导下，独立进行的研究工作及取得的研究成果。尽我所知，除了文中已经注明引用和致谢的地方外，论文中不包含其他人或集体已经发表或撰写的研究成果，也不包含为获得北京语言大学或其他教育机构的学位或证书所使用过的材料。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

签 名：\_\_\_\_\_\_\_\_

日 期：\_\_\_\_\_\_\_\_

学位论文知识产权权属声明

本人郑重声明：本人所呈交论文，是在导师指导下所完成的，论文知识产权归属北京语言大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版本，允许论文被查询和借阅，将论文编入有关数据库进行检索等。本人离校后发表或使用学位论文或与该论文直接相关的学术论文获成果时，署名仍为北京语言大学。

签 名：\_\_\_\_\_\_\_\_

导师签名：\_\_\_\_\_\_\_\_

日 期：\_\_\_\_\_\_\_\_

**目 录**

[摘要： I](#_Toc1502)

[Abstract： II](#_Toc20189)

[第一章 引言 1](#_Toc2748)

[1.1项目背景 1](#_Toc22016)

[1.2 本文使用到的技术 2](#_Toc27966)

[1.2.1 BCC语料库技术 2](#_Toc6996)

[1.2.2词向量技术 3](#_Toc22741)

[1.3论文组织 4](#_Toc8057)

[第二章 研究过程 4](#_Toc2757)

[2.1概述 4](#_Toc5413)

[2.2语法分析 6](#_Toc13165)

[2.2.1分词处理 6](#_Toc31180)

[2.2.2关键词提取 6](#_Toc30388)

[2.2.3停用词过滤 6](#_Toc13535)

[2.3语义分析 7](#_Toc11495)

[2.3.1成语备选集生成 7](#_Toc31143)

[2.3.2计算语义相似度 9](#_Toc20369)

[2.3.3打分排序 9](#_Toc18752)

[第三章 建立系统过程 10](#_Toc6681)

[3.1实验数据 10](#_Toc7502)

[3.2训练模型 10](#_Toc25629)

[3.3系统建设 11](#_Toc18774)

[3.4实验方法 13](#_Toc11243)

[3.5结果分析 14](#_Toc20451)

[第四章 结语与展望 16](#_Toc14727)

[4.1工作总结 16](#_Toc4580)

[4.2工作展望 17](#_Toc18208)

[附录 17](#_Toc12772)

[致 谢 21](#_Toc16863)

[参考文献 22](#_Toc4615)

**基于语义的成语检索方法研究**

**作者：雷舒婷 指导教师：荀恩东**

摘要：

基于语义的成语查询主要是使用户在仅仅知道表达意图而不知道具体成语的情况下能够查询到所需的成语。本文的工作主要有：基于BCC查询引擎启动成语查询服务，对用户以自然语言形式提出的查询请求进行语法分析，提取关键词序列，再由关键词序列查询得到成语备选集。训练词嵌入模型，计算用户请求和备选成语的语义相似度，然后由大到小排序，得到满足用户需求的成语集。经过实验表明，该系统实现了通过语义查询成语的功能，在满足用户此类需求上取得了较好成效。

**关键词**：成语，语义查询，词嵌入，语义相似度

**Research on Semantics-based Idioms Search Method**

**Author: Lei Shuting Tutor: Xun Endong**

Abstract**：**

Semantics-based idioms search is proposed to search the desired idiom, provides idioms that the users only have its meaning which they are expressing.

The main works of this paper are: Starts BCC search engine based on the idiom corpus, extracted keywords by syntax analysis on the query request of users in the form of natural language, and searches alternative idiom set using keywords sequence. Uses inference engine based on description logic for reasoning in idiom area to meet user's searching requirements. The test results indicates that the semantics-based idioms search realizes the function of searching words through semantic and it has a preliminary effect in meeting user's searching requirements.

**Keywords**: Idioms, Semantic search, Word Embedding, Semantic similarity

1. **引言**

**1.1项目背景**

汉语词汇，是汉语里所有词和固定短语的总和。在日常生活中，比如写作时，常常会有这样的情形发生：想要表达某个意思，可是一时之间想不出恰当的词语。而现有的词语（词或短语）查询系统一般是通过词形（写法）和读音（拼音）来查询词语的使用、释义、出处等相关信息。这时，我们上网搜索并不能找到我们需要的词语。例如，在百度上搜索，百度搜索使用的是关键词匹配，可以初步实现对部分用户查询请求进行成语反向查询，但也仅仅只是单纯的返回包含请求中某字的成语，而非与用户查询语义相关的成语（如图1.1中的 “一丘之貉”，“草木皆兵”，“回光返照”等）；成语大全是IT学习网提供的在线的成语词典服务，收录了3万多条成语，提供模糊查找和指定位置查找，但是查询请求最多只能是4个字，不能根据释义来检索成语（如图1.2）；无忧在线成语词典收录了41,843条成语，可以通过对成语的名称、释义、近义词和反义词的检索来查成语，但是根据成语释义检索时，最多只会取输入检索语句的前四个字来进行检索，并没有考虑到用户查询请求的语义信息。

可以看出，此时，现有的基于词形或者读音或者二者结合的方法都无法满足用户的需求，而是需要“基于语义的查询”。在大多数情况下，用户很难只用简单的几个关键词来表达出查询需求。而基于关键词的查询方式主要是对用户输入的查询请求进行字串匹配，通过组合关键词来表达简单的语义信息，这样会结果中会出现很多的无用信息。所以，要实现通过语义来查询词语，基于关键词的查询是远远不够的。针对这一现状，考虑到汉语词汇的丰富性和复杂性，并且成语又是比词的含义更加丰富而语法功能又相当于词的语言单位，所以本文选取成语作为研究对象，探索基于语义的成语查询方法。



图1.1 百度搜索“山峰的成语”结果



图1.2 成语大全搜索“人品质好”结果

**1.2 本文使用到的技术**

1.2.1 BCC语料库技术

本文使用了BCC语料库系统。BCC是北京语言大学大数据与语言教育技术研究所建设的多领域、多语体的全文检索系统。支持字串和词性组合的模式查询，并且可以对查询结果进行统计分析与下载。

BCC语料库支持泛化、模糊、多模态检索，比如支持使用“\*”号实现查询泛化，“.”可以用来进行确定字数的泛化，“[]”可以实现括号内字或词的组合查询。因为实现了对字符的泛化操作，所以BCC系统可以支持长距离依存的语言模式检索，即可以依据句式和词性的结合检索出完整的一句话。这十分有利于对汉语句法句式方面的研究。

1.2.2词向量技术

自然语言理解问题要转化为机器学习问题，第一步就是要把这些符号形式化。NLP中最直观的词表示方法是 One-hot Representation，这种方法把每个词都表示为一个很长的向量，向量的维度是词表的大小，其中绝大多数元素为0，只有一个维度的值为1，这个维度就代表了当前的词，比如：

“话筒”表示为 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 …]

“麦克”表示为 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 …]

但是这种表示存在一个严重的问题，就是“词汇鸿沟”现象：任意两个词之间都是孤立的。因此，Deep Learning中一般用到的词向量并不是刚才提到的用 One-hot Representation 表示的那种很长很长的词向量，而是使用Distributed Representation表示的一种低维实数向量（维度以50维和100维比较常见），如[ 0.792, −0.177, −0.107, 0.109, −0.542, …]。这种表示让相似的词在距离上更近，例如，“麦克”和“话筒”的距离会远远小于“麦克”和“天气”。

Distributed representation最早是由Hinton在1986年提出[1]。其基本思想是通过训练将每个词映射成K维实数向量（K一般为模型中的超参数），通过词之间的距离（比如cosine相似度、欧氏距离等）来判断它们之间的语义相似度。

Word2vec[2]是Google在2013年年中开源的一款将词表征为实数值向量的高效工具，采用的模型有CBOW（Continuous Bag-Of-Words，即连续的词袋模型）和Skip-Gram两种。其使用的便是 Distributed representation的词向量表示方式，其利用深度学习的思想，把对文本内容的处理简化为K维向量空间中的向量运算，计算出向量空间上的相似度。向量空间上的相似度可以用来表示文本语义上的相似度。Word2vec输出的词向量可以被用来做很多NLP相关的工作，比如聚类、找同义词、词性分析等等。

**1.3论文组织**

论文第一章主要介绍了项目背景，研究意义和研究过程中使用到的工具及技术。

第二章主要介绍了具体的研究过程，建立成语检索模型，包括语法分析和语义分析。语法分析又包括分词处理，根据用户输入的自然语言进行语法分析，提取关键词词组，过滤掉停用词后，再根据关键词序列在BCC中检索得到备选成绩集，最后根据语义相似度对备选集进行打分排序，得到检索结果。

第三章主要介绍了成语查询系统的建设过程及结合已有成语词典进行的实验评估过程。

第四章主要是工作总结与展望。

最后是附录，致谢和参考文献。

1. **研究过程**

**2.1概述**

在成语的学习和使用的过程中，很多情况下，用户仅仅知道想要表达的意思，而不知道具体的成语。在现有的成语检索系统上检索无法得到令人满意的结果。例如，如果用户选择通过成语释义来检索成语，分别输入“描写形容品格高尚的人的成语”、“人的品质高尚”和“人品质好” 进行检索，要么是检索不到结果，要么是检索出来的结果完全不同，然而这三句话想要表达的意思是相同的，检索结果也应该相似。出现这一结果的主要原因是，现在的检索系统对用户输入的查询请求进行字串匹配，不会考虑其中的语义关系，也不会进行语义分析。本文针对这一问题探索了基于语义的成语检索方法。图2.1是基于语义的成语检索模型。

语法分析

停用词过滤

关键词提取

分词处理

查询请求

语义分析

排序

查询请求和成语语义相似度计算

否

备选集是

否为空

成语备选集合生成

输出：

满足检索要求的成语

输出：

没有满足要求的成语

图2.1成语检索模型流程图

用户首先以自然语言的方式提出查询请求，该模型会对用户请求进行语法分析，得到关键词序列后，再根据关键词序列在BCC中检索得到备选成语集，然后根据语义相似度对备选成语集进行打分排序，最后得到检索结果。

**2.2语法分析**

用户在进行成语检索时以自然语言的形式输入查询语句，要对这些自然语言所表达的语义信息进行理解，首先就要对用户输入的自然语言进行语法分析。词是最小的能够独立活动的有意义的语言成分。语法分析首先就是要将用户输入的自然语言进行分词处理，然后再从中提取出能够表达句子意义的关键词词组。

在语法分析子系统中，分词处理模块负责对用户输入的查询请求进行分词处理和词性标注，关键词提取模块负责对整个句子的句法结构进行分析，并且结合分词处理后的词性标注提取出关键词，得到相应的关键词集合。

2.2.1分词处理

目前的中文分词和词性标注的算法得到了很好的发展，因此本文没有再开发新的算法，而是使用了上海林原信息科技有限公司的汉语言处理包 HanLp2的分词处理系统进行分词处理和词性标注,主要代码如下所示：

List<String>keywordList=HanLP.extractKeyword(query,5);

对“形容品格高尚的人的成语”进行分词处理和词性标注后的结果为：描写/v,形容/v,品格/n,高尚/a,的/ude1,人/n,的/ude1, 词语/n。其中，v表示动词，n 表示名词，a 表示形容词，ude1表示助词。

2.2.2关键词提取

任何句子都是由关键成分（主语、谓语、宾语等）和修饰成分（定语、状语、补语等）构成的，而提取关键词只需要考虑句子中的关键成分。一般，句子中的主语和宾语大多数是名词和代词，动词和形容词大多数是作谓语。因此，根据语言学知识，可以将名词、动词、形容词作为关键词。如对分词结果：“描写/v,形容/v,品格/n,高尚/a,的/ude1,人/n,的/ude1,词语/n”进行关键词提取，得到一个相应的关键词集合：[品格,词语,高尚,描写,形容]。这样提取的关键词基本上能够反映出句子的核心思想。

2.2.3停用词过滤

信息检索过程中，为了节省存储空间和提高搜索效率，在处理自然语言数据（或者文本）之前或之后会自动过滤掉某些字或词语，这些字或词语即被称为Stop Words（停用词）。

上文提取到的关键词中，描写、形容等是句子的关键成分，但是对于成语检索来说，这些词会引进大量的无关结果，且无实际的意义，应该过滤掉。本文中的类似停用词包括：描写，描述，讲述，表示，表现，形容，相关，类似，成语，短语，词语，愿意，一般，比较，特别，非常等。比如关键词集合[品格，词语，高尚，描写，形容]过滤掉停用词后的结果应该是[品格,高尚]。停用词过滤的主要代码如下所示：

List<String>stopList=Arrays.asList(“描写”，“描述”，“表示”，“相关”，“形容”，“成语”，“词语”，“愿意”，“一般”，“比较”，“特别”，“非常”)；

keywordList.removeAll(stopList);

**2.3语义分析**

2.3.1成语备选集生成

BCC的查询API支持字串和词性组合的模式查询，也支持泛化、模糊、多模态检索，如“\*”号可以泛化查询，表明存在0个或者多个其他字符，“.”可以用来进行确定字数的泛化，“[]”实现了括号内字或词的组合查询等。例如，得到关键词序列[品格,高尚]后，可以向BCC发起查询请求，i\*[品格 高尚]和[品格,高尚]\*i (查询结果如图3.2和3.3所示)，两个请求分别表示品格或者高尚前接的成语和品格或者高尚后接的成语，这样就能够在大规模的语料中查询所有与“品格”和“高尚”共同出现的成语，得到成语备选集。

得到关键词集合后，利用BCC检索得到用户查询备选集，主要代码如下所示：

//发送GET请求

String param = “input=[“+search+”]”\*i&pagenum = 0&pagenum=”+ answerNo+”&sort=1”;

if(keywordList.size()<2){

Param=“input=”+search+”\*i&pagenum=0&pagenum=”+ answerNo+”&sort=1”;

}

String param1 = “input=i\*[“+search+”]i\*[“+

search+”]&pagenum = 0&pagenum+ answerNo+”&sort=1”;

if(keywordList.size()<2){

Param1=“input=i\*”+search+”&pagenum = 0&pagenum=”+ answerNo+”&sort=1”;

}

String response =HttpHepler.sendGet(“

http://202.112.195.124:718/freq”,param);

String responsel = HttpHepler.sendGet(“

http://202.112.195.124:718/freq”,param1”);



图2.2 BCC中检索“i\*[品格 高尚]”的查询结果

图2.3 BCC中检索“[品格 高尚]\*i”的查询结果



2.3.2计算语义相似度

计算查询请求和备选成语集之间的语义相似度的时候，本文充分利用查询请求、成语以及成语对应的释义等信息，形成了相似度计算公式（如公式1所示）。其中，k 表示对用户请求进行语法分析后得到的关键词集合，idiom表示成语，i表示对备选成语释义进行分词处理后得到的关键词序列，2\*f表示的是加分项，f表示的是备选成语是否包含查询请求关键词序列中的字，包含取值为1，反之，取值为0。

distance=max(distance(k,idiom))+max(distance(k,i))+2\*f

k∈keywords list,i∈idiomDef list （1）

2.3.3打分排序

打分完成后，根据得分由大到小排序，将得分最高的20个备选成语反馈给用户。

打分的主要代码如下：

One.m\_nCosMax=vec.disMax(strKeywords,dicts.get(word))

+vec.disMax(strKeywords,word);

排序的主要代码如下所示：

int count=QA\_Answer.length;

for(int i=0;i<count;i++){

for(int j=0;j<count-i-1;j++){

if(((QA\_Word)QA\_Answer[j]).m\_nCosMax<((QA\_Word）QA\_Answer[j+1]).m\_nCosMax){

Object tmp=QA\_Answer[j];

QA\_Answer[j]=QA\_Answer[j+1];

QA\_Answer[j+1]=tmp;

}

}

}

1. **建立系统过程**

**3.1实验数据**

本文的实验数据是从微博、博客语料中筛选得到包含成语的子语料。经过分词处理、词性标注处理后得到了11.8G的文本文件，用于模型训练和系统建设。成语词条来源于《中国成语大辞典》，这是一部大型语文辞书，内容大部分是直接取材于历代的文献，共收录了古今汉语成语20366条，包含成语读音、释义等信息。

**3.2训练模型**

在Linux环境下搭建了word2vec环境，利用成语语料训练窗口大小为5的100维词向量模型（如图3.1所示）。

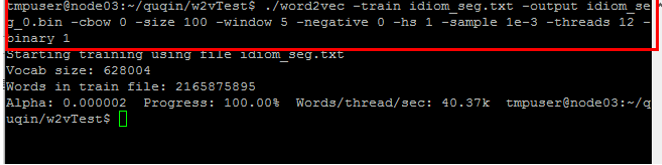


图3.1 word2vec模型训练

**3.3系统建设**

在 Windows 环境下利用PHP和Java建立基于语义的成语查询系统。界面如图3.2所示

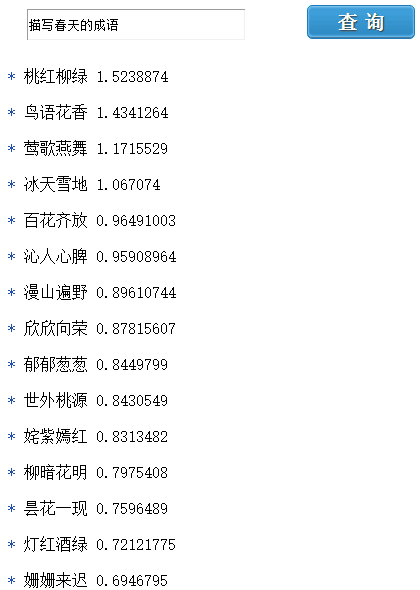


图3.2基于语义的成语查询系统界面

主要的程序代码如下（具体见附录）

<?php

include('corpus\_config.php');

$ciyu="描写春天的成语";

if(!empty($\_REQUEST['ciyu'])){

$ciyu=$\_REQUEST['ciyu'];

}

$search=preg\_replace('/\s/', '', $ciyu);

$cmd="java-jar word2vector.jaridiom\_seg\_0.bin

".$search." 100";

$cmd = iconv("utf-8", "gbk", $cmd);

exec($cmd,$out);

$result=$out[0];

$result = mb\_convert\_encoding($result,"utf-8", "gbk");;

$QA\_Answer=explode("QQ",$result);

?>

然后用word2vec模型计算查询语句和各个备选成语的语义相似度，由大到小排序后，得到满足用户查询要求的成语。例如，输入“描写形容品格高尚的人的成语”，最终检索结果如图3.3所示。



图3.3 本文检索系统对于“描写形容品格高尚的人的成语”的检索结果

**3.4实验方法**

本文的目的是使用户在仅仅知道想要表达的意思而不知道具体成语的情况下能够检索出满足需求的成语。本文的试验都是针对用户的这一需求进行的。向本文的检索系统分别输入“描写形容品格高尚的人的成语”、“人的品质高尚”和“一个品格高尚的人”。检索结果如图3.3,3.4和3.5所示：

图3.4 本文检索系统对于“人的品质高尚”的检索结果



图3.5 本文检索系统对于“一个品格高尚的人”的检索结果

在现有的成语检索系统中输入同样的查询语句，通过与现有检索系统所检索出结果的比较来评测和分析本文方法。

**3.5结果分析**

对于用户输入的检索语句，向本文的检索系统和现有的成语检索系统进行了同样的检索。由于三个查询语句在语义上是相近的，所以其检索结果也应该是语义相近的，从试验结果可以看出来，本文系统检索出的结果基本上相同，从语义上分析能够比较好满足用户的检索需求。而百度搜索不能识别这些特殊的查询请求，没有返回成语查询结果。成语大全，输入的查询语句不能超过4个字，没有根据解释检索成语的功能，无法满足用户的这类检索需求。无忧成语词典，在输入以上三个检索语句后都有检索结果，但是它只是取检索语句的前四个字进行匹配，如果前四个字没有匹配结果就只取前两个字去匹配（如图 3.6所示），因此，虽然都有检索结果，但三个检索结果差异太大，并且，从语义上分析其结果也不能够满足用户的检索需求（具体结果如表3.1所示）。

图3.6无忧在线成语词典“一个品格高尚的人”检索结果



表 3.1四个系统的检索结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 是否有检索结果 | | | 是否有语义分析分析 | 能否从语义上满足用户的检索要求 |
|  | 描写形容品格高尚的人的成语 | 人的品质高尚 | 一个品格高尚的人 |
| 本文检索系统 | √ | √ | √ | √ | √ |
| 百度搜索 | × | × | × | × | × |
| 成语大全 | × | × | × | × | × |
| 无忧在线成语词典 | √ | √ | √ | × | × |

本文做了60次类似的试验，每一次都分别向四个系统输入相同的检索请求进行检索，最后得到的检索成功率是能够从语义上满足用户检索需求的试验次数与总的试验次数的比率，具体结果如表3.2所示。

表3.2 四个系统的检索结果

|  |  |  |  |
| --- | --- | --- | --- |
|  | 有检索结果的次数 | 能从语义上满足用户检索需求的次数 | 检索成功率（总共60次） |
| 本文检索系统 | 60 | 55 | 91.67% |
| 百度搜索 | 20 | 18 | 30% |
| 成语大全 | 0 | 0 | 0 |
| 无忧在线成语词典 | 58 | 29 | 48.30% |

从表3.2的试验结果可以看出来，本文的检索系统从语义上的检索成功率为 91.67%，效果明显的要优于其他成语检索系统。百度搜索能够识别的成语查询请求较少、并且没有语法分析，所以在表3.2中它检索出的能够满足用户需求的成语数量都比较少。表3.2显示，无忧在线成语词典几乎每次检索都会有检索结果，但是它的检索结果很多都不能够满足用户的检索请求，并且还会检索出大量的无关的检索结果。

1. **结语与展望**

**4.1工作总结**

论文主要研究了基于语义的成语检索方法，建设了基于语义的成语检索系统，并且进行了实验。本文系统的目的是，使用户在想要表达某个意思，但是一时之间想不出恰当的词语的情况下能够检索出比较满意的成语。以上的研究结果表明，当用户想通过语义来检索成语的时候，基于关键词的成语检索系统远远不能满足这个需求。主要原因在于，用户的检索需求是语义上的检索，而基于关键词匹配的成语检索仅仅只是对字串的完全匹配，它并没有对用户的检索请求进行语义分析和理解。本文使用了基于大数据的词嵌入模型为核心的成语检索方法，建立检索系统对用户的检索请求进行了分词处理、词性标注和语法分析，并且对成语备选集进行了语义相似度分析，能够较好的满足用户的这种检索需求。

**4.2工作展望**

1.希望以后可以对模型做进一步的改善，添加一些新的功能，如加入成语极性计算，调整语义相似度计算公式的参数等。

2．希望以后可以给系统增加新的功能，如当用户输入词语能够查询到可替代的成语。

3.希望以后可以将这个系统放在为微信平台上或者开发成一个app，方便用户随时使用。

**附录**

主要的程序代码如下

<?php

include('corpus\_config.php');

$ciyu="描写春天的成语";

if(!empty($\_REQUEST['ciyu'])){

$ciyu=$\_REQUEST['ciyu'];

}

$search=preg\_replace('/\s/', '', $ciyu);

$cmd="java -jar word2vector.jar idiom\_seg\_0.bin ".$search." 100";

$cmd = iconv("utf-8", "gbk", $cmd);

exec($cmd,$out);

$result=$out[0];

$result = mb\_convert\_encoding($result,"utf-8", "gbk");;

$QA\_Answer=explode("QQ",$result);

?>

<!DOCTYPE html>

<head>

<meta charset="utf-8"/>

<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />

<title>idiom\_bcc</title>

<style type="text/css">

\*{margin:0px;

padding:0px;

}

.demo\_con{

width: 100%;

margin:0 auto 0;/\*设置上边距40px,左右自动，下边距0\*/

}

.button{

width:20%;

line-height: 32px;

text-align: center;

font-weight: bold;

color: #fff;

text-shadow:1px 1px 1px #333;

border-radius: 5px;

margin:0 10% 10px 10%;

position: relative;

overflow: hidden;

}

.button.blue{ /\*同时包含这两个类名的元素起作用\*/

border:1px solid #1e7db9;

box-shadow: 0 1px 2px #8fcaee inset,0 -1px 0 #497897 inset,0 -2px 3px #8fcaee inset;

background: -webkit-linear-gradient(top,#42a4e0,#2e88c0);

background: -moz-linear-gradient(top,#42a4e0,#2e88c0);

background: linear-gradient(top,#42a4e0,#2e88c0);

}

.blue:hover{

background: -webkit-linear-gradient(top,#70bfef,#4097ce);

background: -moz-linear-gradient(top,#70bfef,#4097ce);

background: linear-gradient(top,#70bfef,#4097ce);

}

.blue:active{

top:1px;

box-shadow: 0 1px 3px #114566 inset,0 3px 0 #fff;

background: -webkit-linear-gradient(top,#1a71a8,#1976b1);

background: -moz-linear-gradient(top,#1a71a8,#1976b1);

background: linear-gradient(top,#1a71a8,#1976b1);

}

.divContent{

postition:absolute;

left:0px;

top:46px;

right:0px;

bottom:0px;

width:100%;

height:100%;

}

.jushou{

color:rgb(34, 79, 174);

}

</style>

</head>

<body>

<!-- 上部的搜索框 -->

<div class="demo\_con" style="margin-top:5px">

<form action="bcc\_idiom.php" method="post" class="formsearch">

<input class="input\_zi" type="text" name="ciyu" value='<?=$ciyu?>' size="29" style="width:40%;height:29px;margin-left:5%;border-width:1px;border-color=black;" />

<input class="button blue" style='font-size:20px' type="submit" value="查 询" style="width:30%;" />

</form>

</div>

<!-- 下边显示的内容 -->

<div class="divContent" id="divContent">

<?php

if(count($QA\_Answer) > 1){

$count=count($QA\_Answer);

$count=$count-1;

for($i=0;$i<$count;$i++)

{

?>

<br/><p>

<span class="jushou">&nbsp;\*&nbsp;</span><?=$QA\_Answer[$i]."<br/>"?>

</p>

<?php

}

}else{

?>

<br/><p>

<span class="jushou">&nbsp;&nbsp;无相应成语</span>

</p>

<?php

}

?>

</div>

</body>

</html>

致 谢

光阴荏苒，岁月如歌，大学生的生活即将结束。回首这四年的生活，我觉得自己收获很多。当然我的进步离不开那些教导过我的老师、帮助我的同学和支持我的家人。对于你们，我充满了感激之情。

　首先我一定要感谢我的导师荀恩东教授。本论文的工作是在我的导师荀恩东教授的悉心指导下完成的，荀教授提出了许多宝贵意见，严谨的治学态度和科学的工作方法给了我极大的帮助和影响，言传身教将使我终生受益。教授不仅悉心指导我完成了论文的工作，在学习上和生活上也都给了我很大的关心和帮助，在此向老师表示衷心的谢意。

在实验工作及论文撰写期间，瞿琴学姐及很多同学对我论文中的研究工作给予了热情的帮助，在此也向他们表达我的感激之情。

感谢我的家人，他们的理解和支持使我能够在学校专心完成我的学业，焉得谖草，言树之背，养育之恩，无以回报，家人永远健康快乐是我最大的心愿。

感谢我的室友们，从遥远的家来到这个陌生的城市里，是你们和我共同维系着彼此之间朋友般的感情，维系着寝室那份家的融洽。四年里发生的事情历历在目，仿佛就在昨天。今后大家就会各奔前程，以后很难会再聚在一起，希望大家珍重。我们在一起的日子，我会永远记得的。

此时，我的心情久久无法平静，从开始进入课题到论文的顺利完成，有许多可敬的师长、同学、朋友给了我无言的帮助，是你们为我撑起一片天空，在这里请接受我诚挚的谢意。今后我会继续努力，以更加积极地态度面对新的挑战。

**参考文献**

[1] Hinton, Geoffrey E. Learning distributed representations of concepts. Proceedings of the eighth annual conference cognitive science society. 1986.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop. ICLR, 2013.

[3] ZangJiaojiao, XunEndong. The Study on Separable Words’ Separable Forms of Modern Chinese. CLSW, 2015.

[4] 饶高琦，荀恩东.大数据视角下的语言实证工具：北语汉语语料库语料库系统 BCC.第十一届 北京市语言学学会年会（北京）. 2014

[5] Mikolov T,Sutskever I,Chen K,et al.Distributed Representations of Words and Phrases and their Compositionality[C].Proceedings of NIPS,2013.

[6] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. Advances in neural information processing systems, 2009:1081~1088.

[7] Zhang Y, Dubrawski A, Schneider J.Learning the semantic correlation: An alternative way to gain from unlabeled text. Advances in Neural Information Processing Systems, 2008:1945~1952.

[8] 邵艳秋、穗志方、吴云芳．基于词汇语义特征的中文语义角色标注研究[J]．中文信息学报， 2009.

[9] 瞿琴、荀恩东、于东．基于语义的成语检索方法研究及试验。第17届汉语词汇语义学国际研讨会.2016.

[10] Turian Joseph, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), 2010.

[11] Eric Huang, Richard Socher, Christopher Manning and Andrew Ng. Improving word representations via global context and multiple word prototypes. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1,2012.