辅助写作中的成语查询

1. 绪论
   1. 研究背景和意义

汉语词汇（中国汉字语言词汇）是汉语里所有词和固定短语的总和，而成语是这些词汇中一部分定型的词组或短句[[1]](#footnote-1)。成语是中国文化的一大特色，它有固定的结构形式和固定的说法，表示一定的意义，在语句中是作为一个整体来使用的；成语作为汉语词汇的重要组成部分，是比词的含义更丰富而语法功能又相当于词的语言单位，存在着丰富的语义关系。

众所周知，成语的重要特点之一就是源远流长，它是我国悠久历史的沉淀。大多数成语的背后都联系着一个形象具体、寓意深长的故事，这些成语故事不仅生动有趣，发人深省，而且富有极强的感染力，更重要的是，以不同形式巧妙利用这些成语故事还可以培养学生的一些相关的写作能力。可以说，成语的正确应用和学生写作能力的培养有十分密切关系。在作文中合理使用成语，可以使作文更有文采、更有美感。现有成语检索系统可通过词形和读音来查询词语的使用、释义等相关信息。然而，常有这样的情形：用户想要表达某个意思，但一时想不出恰当的词语。此时，现有的基于词形或读音或二者结合的方法都无法满足这一需求，而需要“基于语义的查询”。

基于关键词的查询方式可以通过关键词的组合来表达简单的语义信息，但主要是对用户输入的查询请求进行字串匹配，会查询出大量的无用信息，也没有考虑其中的语义关系，例如：用户分别输入“人品质高尚”、“人的品质高尚”和“人品质好”进行检索，检索出来的结果是完全不同的，然而实际上这三句话表达的意思却是相同的。且在大多数情况下用户很难用简单的几个关键词来表达查询需求。所以，要实现通过语义查询词语，基于关键词的查询远远不够。针对这一现状，本课题将探索基于语义的成语查询方法，融合传统的成语查询，形成一个完善的成语查询系统，辅助写作中的成语检索。

* 1. 研究现状
     1. 现有成语查询系统

现有成语查询系统都是通过词形（写法）和读音（拼音）来查询成语的使用、释义等相关信息。

百度搜索使用关键词匹配，初步实现对部分用户查询请求进行成语反向查询，但只是单纯的返回包含query中某字的成语，而非与用户查询语义相关的成语。如图1所示，用户输入“描写山峰的成语”，只是单纯的检索出了词条中包含了“山峰”这一关键词同义词的成语，而非语义相关的，如图中的“一丘之貉”、“开门见山”、“逼上梁山”。

词霸搜索是金山词霸的互联网版本，它提供免费的专业检索服务，是目前市面上最大的互联网多语词典，日均用户量超过100万，用户覆盖量达到3000万之多。词霸搜索包括了200本词典内容，词条数更是达到数百万条之多，覆盖了几十个专业领域，还支持中、英、日、德、法等五国语言的相互查询。然而，对于中文检索部分，它只支持词条检索释义、翻译等信息的正向检索，不能根据释义检索对应成语。

无忧在线成语词典是目前功能最全最强大的成语检索平台，收录了41,843条成语词条，可以通过对成语的名称、释义、近义词和反义词的检索来检索成语，但根据成语释义检索时，最多只截取用户输入检索语句的前四个字来进行字串匹配，并没有真的考虑用户查询请求的语义信息。



图 1 百度搜索“描写山峰的成语”结果

* + 1. BCC语料库技术

本文使用了“北京语言大学语料库中心（BLCU Corpus Center，简称BCC）”提供的语料库系统 。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，并可对查询结果进行统计分析与下载。

BCC设计实现了一种简单的查询语言，即BCC检索式。BCC检索式主要由汉字串（或者词串）、属性符号、通配符、集合符号、离合符号、属性约束符号、空格或“+”组成，支持泛化、模糊、多模态检索[1]。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。本论文中主要用到了BCC检索式的汉字串、属性符号和通配符功能。

1. **汉字串（或者词串）：**

BCC中检索汉字串不需要给出分词信息，例如检索式：“与其说是”，检索包含“与其说是”的实例。

1. **属性符号：**

属性符号是指在标注语料库中，字、词或短语所具有的类型标记。它可以是词性符号、短语类型符号、语义符号等，具体与标注语料库的内容和标注体系相关。目前，BCC中的属性符号主要是指词性符号和短语类型符号，而短语类型符号仅用于具有短语标注的树库中。BCC中汉语语料库采用北京大学的词性体系。例如检索式：“春天\*i”，检索“春天”后接属性符号i（成语）的实例。

1. **离合符号“\*”：**

通常，BCC检索式对应连续的字符串。引入该符号可以描述语言中的各种离合现象。使用该符号的一般形式为：“检索式1\*检索式2”，表示在句子内（对于汉语是小句内），检索符合“检索式1”后接其他成分再接“检索式2”的实例。例如检索式：“春天\* i”是检索“春天”后接其他成分再接成语的情况，检索式：“i\*春天”是检索成语后接其他成分再接“春天”的情况，这两个检索式组合，即可得到语料库中和“春天”共现的所有成语。

* + 1. 词向量技术

自然语言理解问题要转化为机器学习问题，首先需要把这些语言符号（细化到词的粒度）形式化，也就是说，需要找到一种机器能理解的词表示方法。NLP 中最直观、最常用的词表示方法是 One-hot Representation，这种表示方法的基本思想是One-hot编码，它直接以词表大小N作为为向量维度，并把每个词按一定顺序进行编号，然后将每个词表示为一个N维向量，这样表示的词向量只有一个维度的值为 1，这个维度就代表了当前的词的位置，其他维度的值都为 0，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 …]

这种表示方法很直观、很好理解，但是这样表示任意两个词向量之间都是孤立的，无法通过向量值捕捉词之间的相似度关系，就算是近义词甚至是同义词也无法从向量中看出任何共性，这就是常说的“词汇鸿沟”现象。此外这种表示方法还容易引发维数灾难，尤其是在Deep Learning相关应用中，数据量一般都很大。因此，Deep Learning中一般用到的是Distributed Representation表示的一种低维实数向量。

Distributed Representation最早由Hinton于1986年在论文 [2] 中提出，这种表示方法的基本思想是通过训练把每个词都映射为一个 K 维实数向量，这里的K指模型中的超参数，一般取值在50维到100维之间，例如 [ 0.792, −0.177, −0.107, 0.109, −0.542, …] 就是一个分布式表示的词向量。用这种方法表示的词向量，语义相似的词在空间距离上会更接近，例如“渔夫”和“打渔郎”，两者的距离会远远小于“渔夫”和“龙卷风”，因此可以通过词之间的距离（比如 cosine 值、欧氏距离等）来判断它们之间的语义相似度。

* + 1. 词表示技术

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，词表示就是指把自然语言转化为机器能理解的符号表示的过程。近些年，各种基于神经网络语言模型的词表示方法层出不穷，其中最经典的是 2003 年 Bengio 、 Schwenk 等人第一次提出的神经网络语言模型（Neural Network Language Model，NNLM）[3]，在训练语言模型的基础上同时得到了词的表示，该模型采用的是 Distributed Representation 词表示方法，避免了 One-hot Representation 表示的维度灾难问题。2013 年 Mikolov 在神经网络语言模型的基础上进行简化，先后提出了 Skip-gram模型和CBOW模型[4,5,6]，使得词表示更加简单而高效。另外一种主流的词表示方法是基于矩阵分解的技术，其中 Pennington 等人最近提出的 GloVe 模型[7] 同 Skip-gram 和 CBOW 模型一样，被广泛应用于各种自然语言处理任务中。部分科研工作者在它们的基础上进行优化和改造，同样取得了不错的结果。

尹文鹏等利用各种词表示模型构建了一个混合词表示模型[8]。选择的是当前最主流、应用最广的 CW, HLBL, Huang, Word2vec, GloVe五个模型，这样做的优点就是综合各个词表示模型的优势，增加模型的覆盖面，进而增强词表示模型的表示能力。Yu 等借助具有相关语义关系的词语对进行训练[9]，而且该语义关系需要保证一定的质量和数量，通过一个词语的表示来预测另外一个相关词语的表示，称为关系限制模型（Relation Constrained Model, RCM），并将 CRM 和 CBOW 模型进行线性加权的结合。Johansson 等借助一个语义网络的网络结构信息来提升模型的表示能力[10]， 其中，词义的表示由词条的表示和语义网络中的邻居节点的词表示混合而成，而其中的邻居节点也是选择最近的、有直接关系的节点，并保持一定的权重。Chen 等将词义表示应用于词义消歧中，并将词义消歧的结果再用来优化词义的表示[11],他们利用 WordNet 中的注释信息对词义表示进行初始化，其中只利用了注释中的名词、动词、形容词和副词，而且注释中的词和目标词之间的相似度大于一定的阈值才会被选择，选定词的词向量的均值作为词义的初始表示值，然后利用消歧的结果，在 Skip-gram 模型的基础上修改目标函数，得到词义表示。Ling 等认为目标词周围的上下文词语并不是同等重要的，而且主流模型都缺少词序信息，于是在词袋模型的基础上进行扩展，提出了 CBOW with Attention 模型，对上下文中的每个词都赋予一定的权重[12]。Faruqui 等利用外部知识库中的语义关系，借助图模型对词向量进行了翻新和改进[13]。

* 1. 本文工作和创新点
     1. 本文工作

本文将探索基于语义的成语查询方法，并整合传统的正向成语查询，构建一个能全面满足用户查询需求的成语检索系统，使得用户可以在只知道表达意图的时候，检索出合适的成语，也可以在知道确定成语的时候，检索出成语的读音、用法等信息，进而更好地辅助中文写作。具体来说，本文研究内容可分为正向查询和反向查询两大块。

**(1)** **正向查询**

正向查询就是传统意义上的成语查询，实现通过词形（写法）和读音（拼音）来查询成语的使用、释义、背景故事等相关信息。此外，还将基于北京语言大学汉语语料库BCC实现正向模糊查询，例如，查询所有“一…一…”、“三…二…”形式的成语。

**(2) 反向查询**

成语反向查询又可以细分为两种情况，一种是想要查询的成语和查询query是修饰关系，例如“描写春天相关的成语”、“描写春天”检索出“风和日丽”、“阳光明媚”等候选成语，另一种是查询query和需求成语是等价关系，查询query是待查成语的口语表达，例如“心情不好”检索出“唉声叹气”、“闷闷不乐”等成语。

如下图2所示，修饰关系类的反向查询主要分为语法分析和语义分析两个模块，其中，语法分析主要是对查询请求进行分词、词性标注、关键词提取等预处理，得到查询请求的关键词序列，然后进入语义分析，基于查询query的关键词序列通过BCC检索生成备选集，再用Word2vec计算查询query和备选成语的相似度，最后基于相似度打分排序，得到最符合用户检索需求的成语。



图 2修饰关系类的成语检索模型

等价关系的反向查询主要是通过Word2vec和同义词词林对成语解释进行扩展，构建成语和等价口语表达的资源库，以实现用户输入口语表达，就能检索出“同义”成语的需求。

* + 1. 本文创新点

本文工作的创新点包括以下几点：

1. 引入词嵌入技术，通过查询query和备选成语之间的语义相似度来打分排序，实现了基于语义的成语查询。
2. 成语是汉语词汇重要组成部分，是有固定的结构形式和固定的说法的词组或短句，同时也是中国传统文化的一大特色，它代表了一个历史故事或者寓言典故。成语一般都比较短，通常是3到8个字，其中4字成语最为常见，成语的单字一般没有参考意义，但作为整体时有着丰富的含义，因此在语句中成语是作为一个整体使用的。成语和歇后语类似，很大一部分都是直接从古代相承沿用下来的，在用词和构词语法方面不同于现代汉语，两者的构词规则不在一个维度，直接通过词向量来计算查询query关键词集合和成语分词结果之间的相似度意义不大。因此本研究计算语义相似度的时候，充分利用查询query、备选成语以及对应成语解释等信息，而不是单纯计算查询query和备选成语间的相似度，具体相似度计算公式如下：

（公式 1）

1. 将基于语义的成语查询（成语反向查询）细分成了修饰类的查询和等价关系查询，而不是单纯的全部去检索和查询query共现的成语，更好的满足了用户的检索需求。
2. 《中国成语大辞典》是一部大型语文辞书，内容大都直接取材于历代文献，共收录古今汉语成语 20366 条，包含成语读音释义等信息。它是全国优秀畅销书，销售量近二百万册，本文将在其基础上，利用哈工大同义词词林、词向量技术进行语义计算，并结合一定的人工标注，对成语解释进行扩展，构建成语<=>口语表达的资源库。目前还没有这样的资源库，所以这部分工作虽然理论上难点不是很大，但该资源库的构建对后期相关研究很有意义。
   1. 论文组织结构

本文内容主要共分为五个章节：

第一章为绪论部分，介绍辅助写作中的成语查询工作相关的研究背景和研究意义，以及现有相关研究工作的进展，并简述了本文的主要研究内容、研究方法和研究思路，列举出本文采用解决方法的创新之处。

第二章主要介绍本文所采用方法的原理和使用到的知识库。首先是对词表示模型发展史的简单介绍，并重点描述了目前主流的CBOW、Skip-gram和GloVe三个模型的原理。然后介绍了本文用到的知识库，包括现有研究资源，研究技术支持：BCC语料库、google开源工具Word2ec和同义词词林。最后介绍了本文备选成语打分会用到的语义相似度计算，简要介绍了目前主流的应用较广的模型，最后重点介绍了本文选用的VSM模型。

第三章主要是详细描述本文工作的具体内容，分正向查询、等价关系查询和修饰关系查询三个方面来介绍。其中词表示模型的训练和打分排序是每个类型的查询都会用到的，所以单独做了介绍。对于每种查询，都介绍了数据处理、主要流程、备选集生成，以及最终结果排序等内容。

第四章是系统实现与评价部分，介绍了查询效果的评价准则、评价方法、实验所用的数据、本文所用测试集，以及本文研究内容的具体开发实现，包括系统、语言等部分。

第五章是本文的总结和展望部分，总结本文方法中的所取得的成果以及现有方法的不足，并对未来工作进行展望，提出了现有工作可以继续完善的几个方向。

1. 原理和知识库

2.1 词的表示学习

词的表示学习是指通过大规模的训练语料对词语的语法和语义特征进行自动学习的过程。词的表示学习技术一直受到各界研究者的广泛关注，近年来多个词表示的训练模型被陆续提出，并被应用于各项自然语言处理任务中，而且在部分任务中取得了当前最高水平。

2.1.1 词的表示学习的发展

词的表示学习最早是 1986年由深度学习鼻祖Hinton在《Learning Distributed Representations of Concepts》一文中提出的，该文提出了利用神经网络自动生成词的概念表示信息[2]。然而由于当时计算机的计算水平和神经网络的求解算法的限制，以及神经网络对大规模训练数据的依赖，使得词的表示学习并没有得到很好的发展。直到2007年，无监督的受限玻尔兹曼机[16] 加入到神经网络，随着互联网的快速发展海量数据的积累，以及反向传播算法（Backpropagation algorithm, BP算法）[17] 对神经网络算法的调优，大大提高了算法的执行速度。同时硬件方面的发展例如图形处理器（GPU），显著提高了机器的计算能力和效率，使得算法的运行时间明显减少。这些条件使得词的表示学习技术得以迅速发展，并被广泛应用。

到目前为止，几乎所有的词表示模型都是基于一个分布式假设：具有相同上下文的词语同时也会具有相同或相近的语义或语法意义[18]。在此假设之下，人们提出了各种理论对目标词与目标词的上下文之间的关系进行建模。当前主流模型在目标词与目标词的上下文之间的关系，以及目标词的上下文的表示这两方面都不尽相同。当前效果较好、较为常用的词表示模型对比情况如表 1 所示。

表 1 主流词表示模型对比

|  |  |  |
| --- | --- | --- |
| **模型** | **目标词与上下文之间的关系** | **上下文的表示方法** |
| NNLM[3] | 上下文预测目标词 | 合成 |
| Skip-gram[4] | 目标词预测上下文 | 上下文中某个词 |
| CBOW[4] | 上下文预测目标词 | 上下文中所有词的平均 |
| LBL[19] | 上下文预测目标词 | 合成 |
| C&W[20] | 最大化上下文与目标词联合概率 | 合成 |

对于目标词与目标词的上下文之间的关系，表1中的前四种方法都是利用上下文来对目标词进行预测，其使用的是与条件概率p(w/c) 相似的目标函数；C&W模型[20] 的做法是最大化目标词和上下文之间的联合概率。对于目标词的上下文的表示，也是各不相同，或是使用其中一个词进行表示（Skip-gram），或是使用上下文所有词向量的平均值进行表示（CBOW），或者使用上下文中所有词的组合来表示（LBL, NNLM, C&W）。

除了基于神经网络的方法以外，另一种常用方法是基于对上下文的表示矩阵的研究[7,21,22]。该方法用词语用矩阵的行表示，上下文用矩阵的列表示，矩阵中的元素是词语与上下文中的词语的共现频数。

当前最流行、应用最广泛的是CBOW模型、Skip-gram模型和GloVe模型，接下来本文将分别对这几个模型进行详细介绍。

2.1.2 CBOW (Continuous Bag-of-Words Model)

如下图2.1.1所示[23]， CBOW 模型的结构类似于前文提到的神经网络语言模型（NNLM），但是它省略掉了隐含层，是一个两层结构。由下图可以看出，CBOW 模型的思想和标准的词袋模型（BOW）一样，只不过它最终是将词表示为连续向量所以加上了“C(Continuous)”，也就有了人们所说的CBOW模型。CBOW模型没有考虑词序信息，映射层（projection）被所有词共享，所有的词都被映射到同一个位置，对所有词语的表示向量进行累加，然后平均后得到的值作为映射层。丢掉词序信息看起来不太好，会很大程度影响语义表达，然而例如：“研表究明，汉字的序顺并不定一能影阅响读和理解，事证实明当你看这完句话之后才发字现都乱是的”，这句话里好多词顺序都是乱的，然而并不影响对其语义的理解，所以，去掉词序信息并不会太影响语义理解，还能很好的提高模型训练效率。

与NNLM相比，CBOW模型去掉了隐含层，从神经网络结构转变成了log线性结构。由于在计算的时候映射层之后就直接输出目标词是某个词的概率,少了矩阵计算过程，所以大幅度减少了计算时间，提升了模型的训练速度。

|  |
| --- |
| http://images201609.cnblogs.com/blog/1008922/201609/1008922-20160903170047280-734711101.png |

图2.1.1 CBOW 模型结构图

假设一段训练样本为*wt-(n-1),…,wt*, CBOW模型的输入为：

|  |  |  |
| --- | --- | --- |
|  |  | (公式2.1.1) |

模型根据上下文对目标词进行预测：

|  |  |  |
| --- | --- | --- |
|  |  | (公式2.1.2) |

CBOW模型的优化目标为最大化：

|  |  |  |
| --- | --- | --- |
|  |  | (公式2.1.3) |

CBOW模型有两种训练方法，分别是分层Softmax方法和负采样方法。由于算法内容研究不是本文的重点，本文只做简单介绍。

负采样方法 (Negative Sampling, NEG) 是对比估计噪声 (Noise Contrastive Estimation, NCE) 的一个简化版本，用以提高训练速度和改进词向量质量。与Hierarchical Softmax 相比，NEG没有使用复杂的Huffman树，而是选择了相对简单的随机负采样，能大幅度提高性能。该模型每次迭代时都能增加看到的词语的概率，同时降低没看到的词语的概率，所以模型会更快的收敛。

分层Softmax方法通过构造一个树形结构，降低模型求解的时间复杂度。假设有一个由于100个数组成的分布，如果直接通过它们的非归一化概率来计算某个数归一化后的概率，那么需要计算100次。而如果把计算任务进行拆分，将这100个数分成10类，每类10个数，那么在知道这10个类的非归一化概率的前提下，同时知道每类10个数的非归一化概率，此时再来计算某个数的归一化概率，计算量会大大减小，只需要20（10+10）次就能完成，这样一来，通过对计算任务进行分层处理就提高了5倍的计算速度[[2]](#footnote-2)。同样的道理，对计算任务分出更多的层，利用二分法可大大提高效率。该方法就是结合哈夫曼编码，利用高频词的隐含层数目较少，从而减少计算量。输入对应词语的上下文，进行映射，最后模型输出的哈夫曼树，每个词（即树的叶子节点）都有且仅有唯一一条路径可以从树的根结点root被访问到，词被访问到的概率是沿着二叉树路径的所有概率的乘积，最后再进行归一化操作即可。

2.1.3 Skip-gram (Continuous Skip-gram Model)

如图2.1.2所示，Skip-gram模型与CBOW模型一样也是两层结构，但是与CBOW模型利用上下文预测当前词语相反，Skip-gram模型是给定当前词语来预测一定范围内的其他词语，包括出现在当前词语前面和后面的所有词语，也即利用目标词来预测上下文。Skip-gram模型将一个词所在的上下文中的词作为输出，而当前词本身作为输入，也就是说，给出一个词，希望预测可能出现的上下文的词。通过对一个大的语料库进行训练，得到一个从输入层到映射层的权重模型。

一般来说，两个词距离越远，其相关性就越低，所以在模型的训练过程中取样时，会降低离当前词距离较远的词的权重。训练过程中的优化目标为最大化：

|  |  |  |
| --- | --- | --- |
|  |  | (公式2.1.4) |

其中，

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | (公式2.1.5) | |
|  | | |

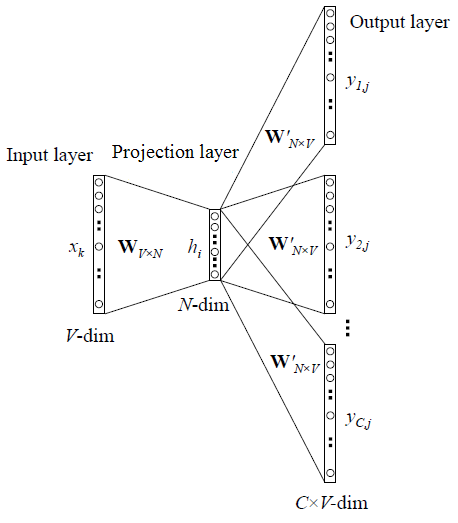


图3.1.2 Skip-gram模型结构图

Skip-gram模型同样有分层Softmax和负采样两种训练方法，前文在描述CBOW模型时已经介绍过，不再赘述。

2.1.4 GloVe (Global Vectors forWord Representation)

与上面提到的CBOW模型和Skip-gram模型基于神经网络的预测机制不同，GloVe模型首先统计语料中词与词之间的共现信息，然后通过矩阵分解的技术得到词的向量表示。虽然很多无监督的词表示学习模型都会用到词语的共现信息，但是如何利用共现信息进行词表示仍然是个问题[7]。

GloVe模型作者通过统计发现，词语之间共现概率的比值更应该被应用于词语的表示学习，具体统计数据如表2.1.2所示。相比于stream，solid和ice共现的概率更高，所以概率的比值*P(k|ice)/ P(k|stream)*较大，而gas与stream共同出现的概率更大，所以它们的概率比值较小。对于water和fashion，与ice和cream的相关性类似，因此概率的比值更接近于1[7]。

表2.1.2 词语的共现概率以及比值信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **概率和比值** | **k = solid** | **k = gas** | **k = water** | **k = fashion** |
| *P(k|ice)* | 1.9\*10-4 | 6.6\*10-5 | 3.0\*10-3 | 1.7\*10-5 |
| *P(k|stream)* | 2.2\*10-5 | 7.8\*10-4 | 2.2\*10-3 | 1.8\*10-5 |
| *P(k|ice)/ P(k|stream)* | 8.9 | 8.5\*10-2 | 1.36 | 0.96 |

2.2 知识库

2.2.1 研究资源

从281.8G（约37亿字）微博、博客语料中筛选得到5.86G包含成语的子语料。经过分词、词性标注处理后得到11.8G文本文件，可用于模型训练和系统建设。选择微博、博客语料而非文学语料，因为文学语料太过书面语化，微博、博客更符合人们的常用语（口语）表达习惯。

成语词条资源方面，本文以《中国成语大辞典》为基础，它是一部大型语文辞书，其内容大部分都是直接取材于历代文献，总共收录了20366多条古今汉语成语，每个成语词条都有读音、释义、背景故事、使用示例等详细信息。而这些成语词条在本文所选语料中平均出现次数高达2284次。

2.2.2 BCC语料库

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，并可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。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[14]。

2.2.3 Word2vec

Word2vec是Google 的 Tomas Mikolov 等人在2013年发表的论文《Efficient Estimation of Word Representations in Vector Space》[4] 中提出的一种把词（中英文均可处理）表征为实数值向量的高效开源工具，其实现代码用C实现，可以便捷的在Linux环境下搭建环境，训练词表示模型。该工具的基本思想是把自然语言中的每一个词，表示成一个统一意义统一维度的短向量，它利用一个单层的神经网络把one-hot形式的词向量映射为Distributed Representation形式的词向量，为了加快训练速度，用了Hierarchical softmax，negative sampling 等trick，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，可以用向量空间中的距离来表示文本在语义上的相似度，而无须知道向量中每个维度的具体意义是什么。Word2vec输出的词向量可以被用来做很多 NLP 相关的工作，比如语义相似度计算、聚类、找同义词、词性分析等等。

Word2Vec 的训练模型，实际上就是具有一个隐含层的神经元网络。Word2Vec的输入是词汇表向量，当看到一个训练样本时，对于样本中的每一个词，就把相应的在词汇表中出现的位置的值置为1，否则置为0。它的输出也是词汇表向量，对于训练样本的标签中的每一个词，就把相应的在词汇表中出现的位置的值置为1，否则置为0。那么，对所有的样本，训练这个神经元网络。收敛之后，将从输入层到隐含层的那些权重，作为每一个词汇表中的词的向量。比如，第一个词的向量是 (w1,1, w1,2, w1,3, ..., w1,m) ，m表示向量的维度。所有虚框中的权重就是所有词的向量值。有了每个词的有限维度的向量，就可以用到其它的应用中。

训练Word2Vec的思想，是利用一个词和它在文本中的上下文的词，这样就省去了人工去标注。Word2Vec 有两种训练模型，即前文提到的CBOW和 Skip-gram，每种模型都有负采样法和分层Softmax两种训练方法，也就是说，Word2Vec一共有4种训练方式供选择。

2.2.4 同义词词林

《同义词词林》是 1983 年梅家驹等人[ 15 ] 编纂而成的，由上海辞书出版社出版，其编写初中是为了提供更多同义词，辅助创作和翻译工作，目前已成为同义词相关研究及词语相似度计算的主要权威性参考。这本词典中不仅包括了一个词语的同义词，例如“家乡”、“故乡”、“出生地”；也包含了一定数量的同类词，即广义上的相关词，例如“状元”、“榜眼”、“探花”。由于《同义词词林》著作时间比较久远，而且之后再没有后续更新，原书中的一些词语随着文化发展变得很不常用，成为生僻词，同时很多新词又没有补充进去，针对这一现状，哈尔滨工业大学信息检索实验室参照多部电子词典资源，投入大量的人力物力资源，按照人民日报语料库中词语的出现频度，只保留出现次数不小于3的词语，完成了一部具有汉语大词表的“哈工大信息检索研究室同义词词林扩展版”。《同义词词林扩展版》共收录词语77343条，比原来的版本多了2万多条。扩展后的《同义词词林》 ，比原始版本包括的词条更多，其含有的语义信息也比之前更为丰富。

同义词词林一共提供了5层编码，第1级用大写英文字母表示；第 2 级用小写英文字母表示；第 3 级用两位十进制整数表示；第4 级用大写英文字母表示；第5 级用两位十进制整数表示。例如，“Gb02B01= 认为 以为 觉得 道 看 当 觉着”，其中“Gb02B01=”是编码，“认为 以为 觉得 道 看 当 觉着”是该类所包含的词语。具体编码如表2.2.1所示。

表2.2.1 同义词词林词语编码表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编码位 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 符号举例 | A | e | 0 | 7 | C | 0 | 1 | =|#|@ |
| 符号性质 | 大类 | 中类 | 小类 | | 词群 | 原子词群 | |  |
| 编码级别 | 第1级 | 第2级 | 第3级 | | 第4级 | 第5级 | |  |

由于第 5 级有的行是同义词，如前面说的“家乡”和“故乡”；有的行是同类词，如“大豆”和“毛豆”；有的行又只有一个词，使用中，有时需要区别对待这 3 种情况，对其进行不同的处理策略，所以有必要再增加标记列来分别表示这几种情形。表 2.2.1 所示，是同义词词林用到的编码表，各编码位按照从左向右的顺序排列，一共有8列，其中第 8 位就是为了区分第5级对应行的词语关系而设计的，该列标记有 3 种取值，分别是“=”、“#”和“@”。其中“=”代表相等，属于同义词，例如“Ca19E01= 冬 冬季 冬天 冬令”，这组词在语义上是可以相互替换的；“#”代表不等，属于同类词（相关词），例如“Af09B05# 状元 榜眼 探花 进士 举人 会元 秀才”，这几个词都是科举考试中的排名，但是它们语义上却没什么相似度，是是同类词而非同义词，在广义上有一定的相关性；“@”表示自我封闭，也即当前行的词在词典中既没有同义词，也没有广义上的相关词，例如“Aa01D01@ 角色”。

2.3 语义相似度计算

2.3.1 语义相似度计算概述

语义相似度计算很多领域都有应用，比如自然语言处理、智能推荐、文本聚类、文本分类、网页判重、信息检索等，它是自然语言研究领域的基础研究课题，是想要理解并比较字串语义的基础，正在被越来越多的学者关注[37]。数字、字符串等值的大小可以直接比较得到，但对于字、词、句子或是文档之间语义相似度比较，就不是那么简单了。相似度计算需要解决两个关键难题，一个是研究对象的特征表示，选取最合适有效的特征来表示研究对象；另一个是特征集合之间的相似关系度量。常见的相似度计算方法有向量空间模型，基于hash方法的相似计算[24,25]，Locality-sensitive hashing[25] 算法族里的代表MinHash方法，以及基于主题的相似度计算等。对于不同的问题域，需要根据数据规模以及时空开销等的限制选择不同的相似度计算方法。本文需要计算的是两个字符串之间的语义相似度，根据问题特性，选用的是应用最广泛的向量空间模型。

向量空间模型（Vector Space Model, VSM）由Salton等人于20世纪70年代提出[27]，它的概念很简单，将每个对象：词、短语、句子或是文档抽象映射为一个带权重的特征向量v = (feature1, w1; feature2, w2; …… ; featuren, wn)，其中feathern表示各个特征，wn表示各个特征对应的权重，这就解决了相似度计算的第一个难题“对象的特征表示”，接下来特征集合间的相似度计算问题就可以简化为向量空间中的向量运算了，以空间距离来表达语义相似度，直观易懂。当语句被表示为语句空间的向量时，就可以通过向量计算来度量语句（或字符串）之间的语义相似度了。VSM模型的相似度比较方法有两种，一种是欧氏距离计算，如公式2.3.1所示。另一种是余弦值计算，由向量间的夹角大小来表示语义相似度。向量空间任意两个向量的夹角在0到180度之间，根据余弦函数曲线知道，在这个范围内余弦值越大就表示向量之间的夹角越小。余弦计算的好处是距离值正好在 [0,1] 区间内，符合相似度百分比的特性，余弦的计算方法为向量内积除以各个向量的模的乘积，具体如公式2.3.2所示。文本处理中最常用的相似性度量方式是余弦距离，因此本文选择的也是余弦距离度量方式。

(公式2.3.1)

(公式2.3.2)

2.3.2 句子语义相似度计算

Palakorn A等人在论文 [28] 中将当前主流的计算句子相似度的算法分为了三类，同时分析了各种方法的原理并对比了各种算法的优势和不足。Li Yuhua等人在论文 [29] 中提出了一种计算词语相似度的方法，并将该算法用在了句子相似度计算的同义词分析上。Donald M等在论文 [30] 中提出了一种计算短语间的相似度的算法。总的来说，现有的句子相似度计算方法，按照其利用的句子信息和句子特征表示的不同可分为两种，一种是基于向量空间模型的方法，这种方法把句子看成是多个词构成的线性序列，只利用组成句子的各个词的词频、词性、长度以及词序等信息来综合衡量语句的语义相似度，而并没有对语句进行语法结构分析，只利用了句子的表层的“字面”信息，没有考虑词之间的相互依赖和影响，没有考虑特定词组合在一起时语义发生的变化，没有考虑句子整体结构的相似性；另一种是对语句进行深度的句法和语义分析，通过深层句法分析找出两个句子的依存关系，并基于依存分析结果计算句子的相似度。

2.3.3 关系向量模型

关系向量模型是殷耀明等人在论文 [31] 中提出的一种同时考虑句子的表层信息和语义分析的句子相似度计算方法。该模型是对Li Yuhua，David M等[32] 提出的算法的改进，通过对待处理序列进行关键词提取，减小待处理序列长度，提高了算法的效率。该模型指出，对于需要计算语义相似度的两个句子（字符串序列），如果两个句子包含了相同的关键词，而且该词在两个句子中的前邻关键词或者后临关键词相同或者是同义词，则表明两个句子不仅有相同的关键词，该关键词在句子中所表示的义项还是同一个，这样加上前后相邻关键词的限定，主要是考虑到多义词的影响，如果满足这一限定，那么可以认为这两个句子相似的概率会更大，也就是说这两个句子的相似度更高，因为这种拥有相同关键词关系反应了句子的局部结构，比单纯的词序信息更能体现句子的语义[31]。模型的相似度计算如公式2.3.1所示，其中Ti和Tj分别表示两个句子的关键词向量，也即关键词集合。

(公式2.3.1)

1. 辅助写作中的成语查询

在成语的学习和使用过程中，很多情况下，与通过成语查询其读音、写法、含义等相反，用户通常只知道想要表达的意思，而不知道具体选用什么成语。这一需求传统的通过成语查询读音、写法等信息的正向检索方式根本无法满足，而目前通过字符串匹配来实现的语义检索也存在很多不足。例如，在现有的成语检索系统中用户选择通过成语释义来检索成语，分别输入“形容人的品格高尚的成语”、“品质高尚的人”和“人品质好”三个查询串进行检索，要么检索不到结果，要么三个查询请求检索出来的结果完全不同，然而这三句话表达的意思是相同的，相应的成语检索结果也应该相似。可见现有的基于语义的成语检索系统并没有真的考虑成语或是查询query的语义信息，也没有对其进行句法分析，而是在资源库中直接对用户输入的查询请求进行字串匹配，并没有考虑其中的语义关系，因此检索出的结果通常都不能满足用户需求。针对这一现状，本文将构建一个能全面满足用户查询需求的成语检索系统，使得用户可以在只知道表达意图的时候，检索出合适的成语，也可以在知道确定成语的时候，检索出成语的读音、用法等信息，进而更好地辅助中文写作。为了实现这一需求，首先根据任务特质，将研究任务分为正向查询和反向查询两部分，其中成语反向查询又可以细分为“修饰关系”查询和“等价关系”查询两种，即最终本文工作主要分为了三部分。

因为两种反向查询均会用到词向量来计算语义相似度，接下来首先介绍本文的词表示模型的训练。

3.1 词表示模型的训练

本文实现基于语义的查询，核心思想就是利用空间量的距离来表示语义上的相似度。而对于空间向量的距离计算，如前面介绍，本文选用的是VSM模型的余弦值计算。词向量技术目前已经比较成熟，并被广泛应用在各个领域。前文对比了很多词向量表示技术和模型，也提到过google开源的词向量工具Word2vec。 Word2vec一经推出便广受欢迎和推崇，一是因为其美妙的向量加减组合运算，例如，，；另一个原因是其高效性，如Mikolov在论文中提到的一个优化的单机版本一天可训练上千亿词[5]。因此，本文训练词向量表示模型的时候就选择了Word2vec。

本文在Linux环境下搭建了Word2vec环境，利用5.86G纯文本成语语料训练窗口大小为5的100维词向量模型，本文选择的是Skip-gram模型的分层Softmax方法来进行训练，其他具体参数设置如图3.3.1所示。

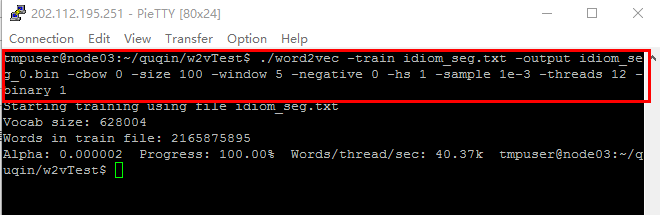


图3.1.1 Word2vec模型训练

3.2 正向查询

3.2.1 正向查询流程

正向查询就是传统意义上的成语查询，实现通过词形（写法）和读音（拼音）来查询成语的使用、释义、背景故事等相关信息。此外，还将基于北京语言大学汉语语料库BCC实现正向模糊查询，例如，查询所有“一…一…”、“三…二…”形式的成语。

如图3.2.1所示，本文首先对用户输入进行判断，然后根据用户输入是否为拼音分别进行处理。用户输入拼音，则直接进行词典检索，得到对应的成语词条以及其释义等信息，例如输入“chunguangmingmei”、 “chūn guāng míng mèi”、 “chun1 guang1 ming2 mei4”等，都能得到词条“春光明媚”相关的信息。

用户输入的是非拼音时，会多一个处理模块“成语答案集生成”。该模块首先对query进行构式分析，根据用户输入的查询请求特征，自动分成两类，一种是需要输入自建BCC检索系统检索得到成语集的，例如“一…一…”、“三…二…的成语”等；另一种是解析处理后直接进行词典检索的，例如“含有动物的成语”、“山开头的成语”、“AABB式的成语”等。



图3.2.1 成语正向查询流程图

3.2.2 数据处理

因为《中国成语大辞典》的词条数有限，并没有涵盖所有的成语，因此本文利用北京语言大学数据库庞大的数据资源，从里面得到更全的成语词表，对成语词条进行补充和扩展。当所查成语不在《中国成语大辞典》原始词条中时，也就是说自封闭的系统无法提供词条的详细信息时，利用百度汉语，抓取确定成语的读音、释义等信息，然后完善本文用到的成语词典。

为了满足拼音检索的需求，支持多种拼音输入形式，需要对成语词典进行预处理，每个词条都有两个拼音字段DislpayPY，最后呈现给用户的拼音，例如“chūn guāng míng mèi”，另一个是PY，用来匹配的字段，例如“chunguangmingmei”。

“含有动物的成语”这类需求，经过网上调研及对成语词条的分析，最后确定了四个类似的分类“动物”、“植物”、“天气”和“生肖”。这四类具体包含关键字如下表所示。

表3.1.1 正向查询特殊类别词表

|  |  |
| --- | --- |
| 类别 | 词表 |
| 动物 | 鱼、鸟、狼、鹤、蜂、鹰、鹿、猫、马、龙、虎、牛、狗、鸡、鼠、蛇、羊、兔、猴、猪 |
| 植物 | 花、树、草、桃、梅、菊 |
| 天气 | 风、雨、火、冰、电、雷、雪、寒、冷、暖、热 |
| 生肖 | 马、龙、虎、牛、狗、鸡、鼠、蛇、羊、兔、猴、猪 |

3.3 等价关系反向查询

等价关系成语反向查询，以下简称等价类类查询，是指所检索得到的成语是用来表示与查询query相同或相近意思的，例如“心情不好”、“形容心情不佳”检索出的 “唉声叹气”、“闷闷不乐”等成语，这样的成语和查询query主体“心情不好”是“等价”关系。等价类查询的实现，主要是通过向量计算和同义词词林对成语解释进行扩展，构建成语和等价口语表达的资源库，以实现用户输入口语表达，就能检索出所需成语。

3.3.1 人工扩展

《中国成语大辞典》中成语的释义大多是比较完整的语句，而且通常还伴有成语的出处或者简要的背景故事等信息，例如，成语“唉声叹气”原始的解释为：*因伤感、烦闷或痛苦而发出叹息声。明·凌濛初《二刻拍案惊奇》卷三十八：“终日价没心没想，哀声叹气”* [38]。很明显，用户在查询成语的时候是不会这么说的，而且直接用这样的释义串去与用户输入的查询query计算向量距离，比较语义相似度也没有意义，因此，本文首先对成语解释进行分词处理，然后人工过滤，去掉原始释义中的无关信息，并把其中的书面语改成口语，比如把“貌”、“状”改成“的样子”，得到人工扩展后的一次扩展集。

具体来说，人工扩展的工作主要包括以下四个方面：

1. 去掉无关信息

这一步处理主要是去掉成语原始释义中的出处、使用例句、背景故事等信息，以及其中的“描写”、“形容”、“比喻”、“暗喻”等无实际意义的词，例如前面说的“唉声叹气”，去掉其释义中的背景故事信息后，得到“因伤感、烦闷或痛苦而发出叹息声”；又例如成语“心如刀割”原始的释义为“*比喻伤心痛苦之极。*”，去掉无关信息“比喻”后得到简化表达“伤心痛苦之极”。这里“比喻”这样的成分在本文研究中被视为无意义词，会当作停用词被过滤，本文研究中类似的停用词一共有62个，具体如下表3.3.1所示。

表3.3.1 停用词表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 形容 | 泛指 | 指 | 表示 | 比喻 | 喻 | 谓 | 犹言 |
| 后多指 | 多指 | 现也指 | 喻指 | 现喻指 | 亦比喻 | 多形容 | 常形容 |
| 意谓 | 亦以 | 用以称 | 后以指 | 后也比喻 | 言即 | 常喻 | 指代 |
| 用作 | 常以比喻 | 称美 | 后因以比喻 | 亦用为 | 本指 | 后用以表示 | 泛称 |
| 后亦以比喻 | 借指 | 借以 | 赞美 | 借以赞美 | 引申为 | 今多用于指 | 本谓 |
| 多用于 | 旧指 | 多用以 | 称颂 | 颂扬 | 代称 | 旧谓 | 后亦指 |
| 极喻 | 常用为 | 现解为 | 用以 | 人的 | 后用为 | 极言 | 后亦用以 |
| 亦用以称美 | 表现 | 称誉 | 用来表达 | 亦用于指 | 表达 |  |  |

1. 书面语改口语

这一步主要是把书面语改成口语化的表达，例如成语“爆跳如雷”原始解释为，“盛怒貌。”，经过这一步的修改扩展后，得到的表达有“盛怒的样子”、“特别生气的样子”等。

1. 释义中多个并列成分拆分组合

还是以成语“唉声叹气”为例，它的原始释义（因伤感、烦闷或痛苦而发出叹息声）中有多个表达相同或相近意思的修饰成分并列使用，然而用户在查询的时候一般情况下不会这么完整的输入这几个方面的意思，所以这一步会针对这种情况进行拆分和组合，得到更多的表达形式。下图3.4.1所示，就是成语“唉声叹气”的释义拆分组合后的结果。

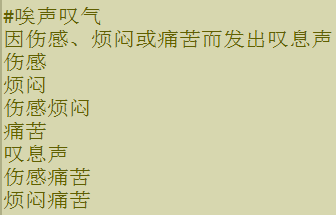


图3.3.1 “唉声叹气”释义人工扩展结果图

1. 句式丰富

成语的原始释义中句式单一，而后期的词向量计算扩展和同义词词林扩展都只能在词汇层面扩展，因此先要人工丰富表达句式，比如成语“心如刀割”的解释“伤心痛苦之极”，可以增加句式“特别心痛”，即倒装主体“心痛”和程度副词的顺序，增强解释的表达能力，类似的例子还有很多，此处不一一赘述。

3.3.2 词向量计算扩展

3.2中利用本文研究语料训练了词向量模型，前文也提到可以利用向量距离来表征词汇或是语句的语义相似度。词向量计算扩展要做的是在人工扩展的基础上，对成语的口语表达进行分词处理，然后通过Word2vec计算和Perl脚本处理等步骤，对口语表达进行扩展。具体工作分以下几个部分：

(1) 分词3.1.1中人工扩展后的成语词典结果，并统计分词结果中字数大于1的词的个数，如图3.3.1所示，为部分过滤单字词后保留下来的结果。



图3.3.1 分词过滤后结果示意图

(2) 对于步骤(1)中得到的每个词，利用Word2vec计算其语义最相近词的top20。例如，词语“伤心”词向量计算得到的语义相近词有“委屈”、“失望”等，具体如图3.3.2所示。



图3.3.2 “伤心”的Word2vec扩展结果

(3) 目前所有词表示模型都是基于一个分布式假设：具有相同上下文的词语同时也会具有相同或相近的语义或语法意义。因此实际上步骤 (2) 扩展出来的是和“伤心”拥有相似上下文的词，而非完全意义上的相同或相似，例如近距离词中的“生病”、“害怕”等。为了尽可能避免这一假设对词汇扩展的影响，这一步会对扩展词再次进行分词处理，然后只保留和原始词词性相同的词。如图3.3.3所示，这里原始词“伤心”的词性是a（形容词），去掉非形容词的扩展词最后保留下来的词只有“失望”、“悲伤”、“痛苦”、“难受”、“难过”、“心酸”、“绝望”，如图3.3.4所示。可以看出，这一策略很好的拒识掉了那些语义上不那么相近但上下文又比较相似的词。

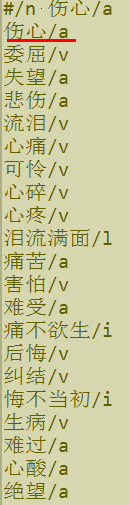
 

图3.3.3 扩展词分词结果图 图3.3.4“伤心”保留扩展词结果

(4) 这一步要做的是利用Perl脚本对原有释义中多于1个字的词，一一用步骤(2)中得到的近义词替换，得到更多的口语表达形式。例如，如图3.3.5所示，成语“抱头大哭”有两个口语表达，其中一个是“非常伤心”，那么现在会一一用“非常”和“伤心”通过Word2vec计算出的近义词来替换，得到扩展后的口语表达，如表3.3.1所示，通过对“非常”和“伤心”进行同义词替换后，最后得到了65个口语表达，比较全面的覆盖了用户表达“非常伤心”的时候的各种说法。

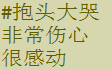


图3.3.5 “抱头痛哭”口语表达示意图

表3.3.1 “非常伤心”词向量计算扩展表达结果

|  |
| --- |
| 比较 伤心 |
| 特别 伤心 |
| 很 伤心 |
| 极为 伤心 |
| 更为 伤心 |
| 更加 伤心 |
| 尤为 伤心 |
| 极其 伤心 |
| 极 伤心 |
| 非常 失望 |
| 非常 悲伤 |
| 非常 痛苦 |
| 非常 难受 |
| 非常 难过 |
| 非常 心酸 |
| 非常 绝望 |
| 比较 失望 |
| 比较 悲伤 |
| 比较 痛苦 |
| 比较 难受 |
| 比较 难过 |
| 比较 心酸 |
| 比较 绝望 |
| 特别 失望 |
| 特别 悲伤 |
| 特别 痛苦 |
| 特别 难受 |
| 特别 难过 |
| 特别 心酸 |
| 特别 绝望 |
| 很 失望 |
| 很 悲伤 |
| 很 痛苦 |
| 很 难受 |
| 很 难过 |
| 很 心酸 |
| 很 绝望 |
| 极为 失望 |
| 极为 悲伤 |
| 极为 痛苦 |
| 极为 难受 |
| 极为 难过 |
| 极为 心酸 |
| 极为 绝望 |
| 尤为 失望 |
| 尤为 悲伤 |
| 尤为 痛苦 |
| 尤为 难受 |
| 尤为 难过 |
| 尤为 心酸 |
| 尤为 绝望 |
| 极其 失望 |
| 极其 悲伤 |
| 极其 痛苦 |
| 极其 难受 |
| 极其 难过 |
| 极其 心酸 |
| 极其 绝望 |
| 极 失望 |
| 极 悲伤 |
| 极 痛苦 |
| 极 难受 |
| 极 难过 |
| 极 心酸 |
| 极 绝望 |

3.3.3 同义词词林扩展

3.3.2中通过词向量计算，选择了top20来筛选，先增大召回再通过后期策略来过滤不合理的近距离词。在词向量计算扩展结果基础上，若再用同义词词林计算一次词语的相似度，再扩展一次近距离词，这样通过词向量计算的语义相似度和基于同义词词林计算的词语相似度误差累加，会放大本系统的表达不可信性。因此，这里不会在利用同义词词林计算一次词语相似度，而是利用同义词词林的编码特点，前文介绍《同义词词林》编码规则时提到过，第8位为“=”时，表示的是相等，也就是说那一组词为同义词。这一步扩展会对编码第8位为“=”的那些词进行替换，例如同义词词林中有一组词是“Ga01B01= 悲伤 伤悲 伤心 伤感 熬心 难过 难受 ……”，那么会把“悲伤”、“伤悲”、“伤心”这一组词互相替换，生成更多的表达。

需要说明的是，鉴于初始版本的《同义词词林》词条数量较小，且是1983年编写的，很多词现在已经不常用或是含义发生了变化，本文选用的是《哈工大信息检索研究中心同义词词林扩展版》。

* 1. 修饰关系反向查询

修饰关系成语反向查询，以下简称修饰类查询，是指所检索得到成语是用来形容、修饰查询query关键词的，例如“描写春天相关的成语”、“描写春天”检索出的“风和日丽”、“阳光明媚”经常被用来描写“春天”，这样的成语和查询query主体“春天”是共现关系。

3.4.1 修饰类查询流程

如图3.4.1所示，修饰系查询主要分为语法分析和语义分析两个模块，其中，语法分析主要是对查询请求进行分词、词性标注、关键词提取等预处理，得到查询请求的关键词序列，然后进入语义分析，基于查询query的关键词序列通过BCC检索生成备选集，再通过计算词向量间的距离来表示查询query和备选成语的语义相似度，最后基于相似度打分排序，得到最符合用户检索需求的成语。



图3.4.1 修饰关系类的成语检索模型

3.4.2 查询query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，要理解这些自然语言所表达的语义信息首先要对用户输入的自然语言进行语法分析[39]。词是最小的能够独立使用的有意义的语言单元，因此语法分析模块首先需要将用户输入的查询query进行分词处理，然后再利用汉语语法知识、句法特征，从中提取出能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（图3.4.1所示）将对用户输入的查询请求进行分词和词性标注；关键词提取模块对整个句子的句法结构进行分析，并结合分词模块输出的词性标注结果提取出关键词，最后再过滤停用词得到输入序列对应的关键词集合。

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HanLp[[3]](#footnote-3)的分词处理系统进行分词处理和词性标注。例如，输入“形容人的品格高尚的成语”，得到的的词性标注后结果为“形容/v 人/n 的/ude1 品格/n 高尚/a 的/ude1成语/n”。其中，词性标注词性标注规范采用的是采用的是北京大学标准，v表示动词，n表示名词，a表示形容词，ude1表示助词。

1. **关键词提取**

汉语中的句子都是由关键成分与修饰成分构成的[31]。关键成分（主、谓、宾等）对句子起到主要作用，能够很大程度上代表句子所表达的意思，修饰成分（定、状、补等）对句子起次要作用，主要是修饰关键成分或者强调程度，因此提取关键词时只需要考虑句子中的关键成分，在进行句子相似度计算的时候也只需要考虑关键成分即可。根据语言学知识，通常情况下，现代汉语句子中的主语主要由名词和代词构成；谓语多为动词或者是形容词；宾语也主要是名词或者代词。因此，可以将句子中的名词、动词、形容词和副词作为关键词，并在计算句子相似度的时候只考虑这些关键词，这样既可以保证计算的准确率，又可以大大提高计算效率。例如，对于分词和词性标注（本文使用的是北京大学词性标注标准）结果：“描写/v形容/v 人/n 的/u 品格/n 高尚/a 的/u成语/n”进行关键词提取，得到相应的关键词集合：[品格, 成语, 高尚, 描写, 形容]。这样提取的关键词具有一定的句法结构信息表达能力，基本上能反映出句子的核心思想，还过滤掉了无关词汇信息，减少了后续处理和计算的复杂度。

3.4.3 停用词过滤

在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言数据（或文本）之前或之后会自动过滤掉某些字或词，这些字或词即被称为Stop Words（停用词[[4]](#footnote-4)）。停用词主要分为两类，一种是使用很频繁的词，例如“你”、“我”、“他”等，另一种是在文本中出现频率很高，却没有实际意义的词，例如“在”、“的”等。

不同的任务，需要过滤的停用词不同，对于本文的成语检索任务，前文提取的关键词“描写”、“形容”等是句子的关键成分，然而对于成语检索这一特定问题，这些词不仅没有实际意义，还会引进大量无关结果，影响用户查询体验，所以在把提取的关键词序列输入BCC检索生成备选集之前，应该首先过滤掉没有意义的“句子关键成分词汇”，例如关键词集合 [品格, 词语, 高尚, 描写, 形容] 过滤停用词后得到的真正用来生成备选集的序列是 [品格,高尚]。前文讲述构造“成语<=>口语表达”资源库的时候，在原始词典资源中得到了62个成语领域没有实际意义的词汇，在此基础整理的出25个“描写”这样停用词（如下表3.4.1所示），以及34个用来修饰这些停用词的词汇，如下表3.4.2所示。在基本停用词表3.4.1的基础上，通过词向量计算扩展每个词距离最近的top10，然后对计算结果进行人工过滤，最后得到本文所用的含有XX个词的完整体用词表，如表3.4.3所示。

表3.4.1 query处理停用词表 (1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 形容 | 泛指 | 指 | 表示 | 比喻 | 喻指 | 称 | 指代 | 称美 |
| 借指 | 赞美 | 称颂 | 颂扬 | 代称 | 极言 | 表示 | 表现 | 称誉 |
| 表达 | 描写 | 描述 | 相关 | 类似 | 成语 | 词语 |  |  |

表3.4.2 停用词修饰词汇表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 多 | 常 | 经常 | 通常 | 用以 | 用作 | 用为 | 借以 | 用于 |
| 用来 | 多用以 | 多用作 | 多用为 | 多借以 | 多用于 | 多用来 | 常用以 | 常用作 |
| 常用为 | 常借以 | 常用于 | 常用来 | 通常用以 | 通常用作 | 通常用为 | 通常借以 | 通常用于 |
| 通常用来 | 经常用以 | 经常用作 | 经常用为 | 经常借以 | 经常用于 | 经常用来 |  |  |

3.4.4 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，例如“\*”号实现查询泛化，表示中间存在0个或多个其他字符，“[]”实现了括号内字或词的组合查询。例如，得到关键词序列“[品格,高尚]”后，可向BCC发起查询请求，“i\*[品格 高尚]”和“[品格,高尚]\*i”，两个请求分别表示品格或高尚前接成语和品格或高尚后接成语，这样就能在大规模语料中查询所有与“品格”和“高尚”共现的成语，得到备选集。

3.5 成语极性判断

成语是有情感极性的，例如“唉声叹气”和“兴高采烈”都是形容心情的成语，但是一个是表示“心情不好”，一个是表示“心情好”，在用户检索“心情不好”的时候除了“兴高采烈”是不满足检索需求，且会严重影响检索体验的，因此对于成语还需要进行极性判断。

前面介绍过本文选用的研究语料中包含微博语料，其中包含了微博表情信息，而微博表符号是有特殊符号标记的，例如“[开心]”，本文利用这些表情符号信息来计算成语词条的极性。首先整理出所有微博语料中的情感符号，并人工过滤，最后保留了1123条，然后综合了多个数据堂的情感词典，具体包括CASIA汉语情感语料库[[5]](#footnote-5)、中文情感极性词典 NTUSD [[6]](#footnote-6)、NLP&CC 2013 中文微博情绪识别样例数据[[7]](#footnote-7)，来判断这些情感符号的极性，最后因为成语和这些情感符号共现，出现在同一自然句中，那么可以认为这些情感符号的极性即代表了句子的极性，也就可以代表对应成语的极性了。

3.6 打分排序

在计算查询请求和备选成语之间的语义相似度（即距离distance）的时候，本文充分利用了查询请求、成语以及成语对应的释义等信息，形成了最终的相似度计算公式（公式3.6.1所示）。其中，k表示对用户查询请求进行语法分析得到的关键词集合；i表示对备选成语的释义进行分词处理及语法分析后得到的关键词序列；I表示备选成语；\*f是加分项，f表示备选成语是否包含查询请求关键词序列中的字，如果包含则f取值为1，反之，取值为0，为加分项的权重，经过多次实验，发现 =2 时效果最好。打分完成后，根据得分由大到小排序，并取得分最高的20个备选成语反馈给用户。

(公式3.6.1)

对于具体的两个序列的相似度计算，例如，本文选用的是关系向量模型，不同的是殷耀明等提出的关系向量模型在计算词之间相似度的时候用的是《知网》 ，而本文已经有训练好的词向量模型，因此本文在原有模型基础上做了改进，通过Word2vec计算词的向量距离来表示词的相似度。

1. 系统实现与评价

4.1 实验数据

从281.8G微博、博客语料中筛选得到5.86G包含成语的子语料。经过分词、词性标注处理后得到11.8G文本文件，用于模型训练和系统建设。

成语词条方面，本系统中初始资源是《中国成语大辞典》 ，它是一部大型语文辞书，内容大都直接取材于历代文献，共收录了古今汉语成语 20366 条，包含成语读音、释义等信息。而这些成语词条在本文所选语料中平均出现次数高达2284次。

* 1. 系统实现

本文工作的核心是封装了一个Jar的检索功能包，可以用不同语言，在不同环境下调用，为了便于展示，方便用户测试和评价检索功能效果，本文选择的是通过网页来呈现检索效果。本文实现成语检索的框架如图4.2.1所示，首先从浏览器得到用户输入，然后传给构式分析模块，根据用户查询请求确定到底调用哪种成语检索接口，生成所需检索到的成语答案集，最后再返回给浏览器端，直观地反馈给用户。本文的浏览器端代码比较简单，只是用于直观的展示检索结果，没有把重心放在一个酷炫网站的开发上，网站代码是用PHP实现的，如图4.2.2所示，PHP负责得到用户输入，然后通过命令行调用jar包得到检索答案。构式分析部分是在JAVA环境下实现的，主要是通过分析用户输入的检索串形式，来确定调用正向查询、等价类查询亦或是修饰类查询，最后根据构式分析的结果调用对应的检索接口生成答案返回给浏览器端，第3章已经介绍过，等价类的查询和正向查询都是直接通过词典检索得到答案的。



图4.2.1 本文检索系统架构

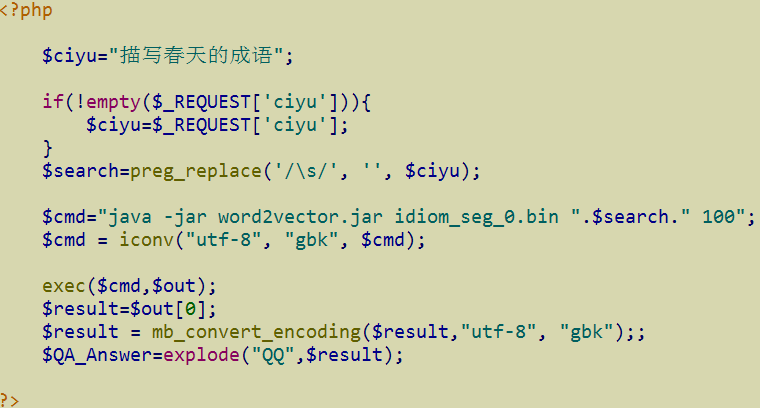


图4.2.2 浏览器端代码示意图

4.3 修饰类查询实现

修饰类的检索比较特殊，会先通过BCC检索得到备选成语集，然后通过打分排序得到答案集，如图4.3.1所示，修饰类查询首先会对query进行关键词提取以及过滤停用词的操作，然后通过URL传参的形式由系统搭建的BCC检索服务得到所有与查询query关键词组合共现的成语。如图4.3.2，BCC检索得到的结果包含了共现成语和对应的关键词以及对应组合在语料库中的频数等信息，例如“姹紫嫣红\*春天 207”表示在语料中“春天”前面出现“姹紫嫣红”的次数为207。函数*public List<QA\_Word> getAnswers(String modelPath, String content, String content1, String strKeywords, String query)* 负责对图4.3.2中的结果进行解析，得到所有备选成语和对应频数信息，最后通过打分排序，得到答案集。



图4.3.1 修饰类查询备选集生成代码示意图

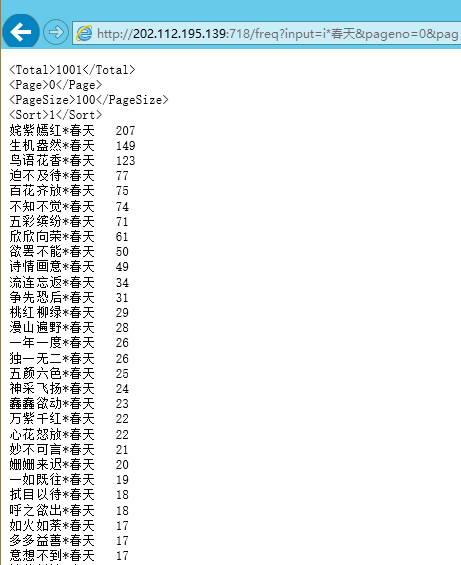


图4.3.2 BCC检索结果示意图

* 1. 实验评价

由于之前尚未有完全类似的本文相关的研究工作，没有一个已知的特定评测集来验证检索效果，因此本文分别通过本文系统自评来对系统进行验证，同时还通过和现有其他成语检索平台进行互评，来说明本文方法的有效性。

* + 1. 本文系统自评

系统自评阶段，首先随机找了5个人，每人提供20条对于这样的平台他会想到的查询请求，部分查询query如图4.4.1所示，对5人提供的100条query去重，最终剩下87条。然后系统对这87条query一一进行检索，得到成语结果集，对于如何评价这些结果方面，本文选则了“众测”的方式。问卷星是一个专业的在线问卷调查、测评、投票平台，专注于为用户提供功能强大、人性化的在线设计问卷、采集数据、自定义报表、调查结果分析系列服务。笔者在通过问卷星发布了评估任务，对于每一个query的检索结果设置了“完全满足检索需求”、“没有想要的成语”、“既有想要的成语又有不相关成语”、 “完全不满足检索需求”四个单选项，每个等级的具体评价标准为，“完全满足检索需求”：检索结果中有合适的成语且其他成语和检索请求语义相关；“没有想要的成语”，结果成语和检索请求语义相关，但是没有完全满足检索需求的成语；“既有想要的成语又有不相关成语”：检索结果中有需要的成语，但是有部分和检索需求不相关成语；“完全满足检索需求”：检索结果中没有用户所需成语，同时还包含有和检索需求语义不相关的成语。本文共邀请100个被试参与了调查，也即是本文共测试了870次（），最终的评估结果如表4.4.1所示，由表中结果可以看出，完全满足检索需求的有63.11%，既有所需成语又有不相关成语占比23.44%，也就是说本文检索系统能满足用户检索需求的概率达86.55%，较好满足了用户查询需求，是该领域的一次不错的尝试。

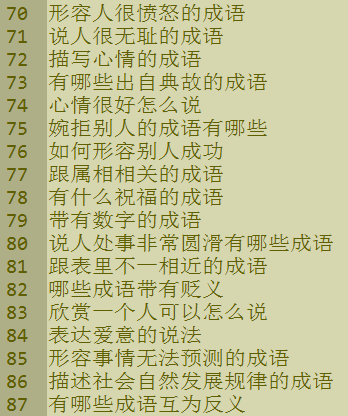


图4.4.1 测试query示意图

表4.4.1 问卷星调查结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 完全满足检索需求 | 没有想要的成语 | 既有想要的成语又有不相关成语 | 完全不满足检索需求 |
| query条数 | 549 | 65 | 204 | 52 |
| query占比 | 63.11% | 7.47% | 23.44% | 5.98% |

4.4.2 现有平台互评

本文的主要目的是基于语义的成语查询，使得用户在明确表达意图而不知道具体成语的情况下能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。 向本文检索系统分别输入“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 4.4.2和4.4.3所示：

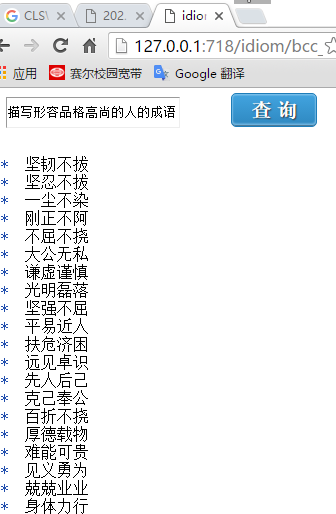
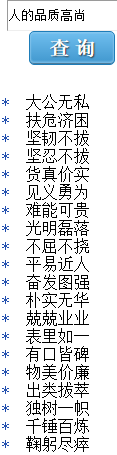
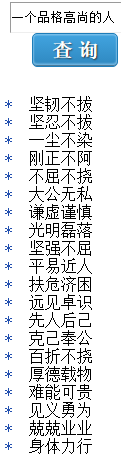
  

图4.4.2 系统检索界面 图4.4.3 本文系统检索结果

系统效果自评有一定的局限性，为了更客观的评估系统，本文还另外选取了3个现有成语检索系统（百度搜索、ICIBA成语词典、无忧在线成语词典）进行互评，通过输入相同检索串，然后与现有检索系统所检索出结果进行对比来评测和分析本文方法的有效性。

“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”三个输入从字面上和句法结构来看，其表达形式各有不同，但是在语义上三者却是相近的，要表达的核心思想都是“人 品格 高尚”，所以这三个查询请求检索得到结果也应该是相同或者相似的。从试验结果可以看出，本文系统检索出的结果基本相同，从语义上分析能较好满足用户的检索需求。百度搜索没有反向查询功能，不能识别这些特殊查询请求，没返回成语查询结果；ICIBA 成语词典广受欢迎主要是因为其丰富的资源库，以及强大的正向查询功能，但是它没有根据解释检索成语的功能，无法满足用户的这类反向检索需求；无忧在线成语词典支持简单的基于子串匹配的语义检索，用户可以根据成语释义进行逆向检索，对于以上三个查询请求都有检索结果。如图 4.4.4所示，当检索串太长时，它只取检索语句的前四个字进行字符串匹配，如果前四个字没有结果，再退化处理，选取前两个字去匹配。如表4.4.2所示，虽然无忧在线成语词典三个查询query都有检索结果，但三个检索结果差异太大，大多是不相关成语，从语义上分析其结果也没有与用户查询请求语义相近的成语，不能真正满足用户“基于语义”的检索需求。



图4.4.4 无忧在线成语词典“描写形容品格高尚的人的成语”检索结果

表4.4.2 试验结果1

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

本文从前文提到的87查询请求中随机选择了60条，进行了类似的试验，每一次都分别向四个系统输入相同的查询query，最终的实验结果如表4.4.3所示，其中检索成功率是指能从语义上满足用户检索需求（即检索结果中包含用户需要的成语）的试验次数与总的试验次数的百分比。

表4.4.3 试验结果2

|  |  |  |  |
| --- | --- | --- | --- |
|  | 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% |
| ICIBA成语词典 | 0 | 0 | 0 |
| 无忧在线成语词典 | 58 | 29 | 48.30% |

从表 4.4.3 的试验结果可以看出，本文系统语义上的检索成功率高达 91.67%，效果明显的优于现有的其他成语检索系统的水平（48.3%）。百度搜索能识别的成语查询请求较少，如表4.4.3所示仅有20次，而且其不支持对检索请求进行语法分析，所以在表 4.4.3 中它检索出的能从语义上满足用户需求的成语数量也很少，只有18条。无忧在线成语词典有58次都有检索结果，几乎每次检索都有成语结果返回，但是它的根据释义查询只是单纯的字符串匹配，而且最多只能取输入query的前4个字进行检索，因此其检索结果很多都不能满足用户的检索需求，并且还会检索出大量无关的结果。

1. 总结和展望

本文研究的目的是探索基于语义的成语查询方法，并融合传统的正向成语查询，细化用户查询的意图，分三种不同查询来构建一个能全面满足用户查询需求的成语检索系统。用户可以在知道确定成语的时候，检索出成语的读音、用法等信息，也可以在只有表达意图而不知道具体成语的情况下能检索出比较合适的成语，全面辅助写作中的成语查询。传统的基于关键词的成语检索系统远不能满足语义检索成语这个需求。其根本原因在于，通过表达意图查询对应成语需要的是语义上的检索，而传统的基于关键词匹配的成语检索方式只是对字串的进行完全硬匹配，并没有对用户输入的查询query进行语义分析。本文使用基于大数据的词嵌入模型为核心的成语检索方法，搭建系统对用户的检索请求进行了分词、词性标注和语法分析，并通过词向量计算对备选集成语和查询query进行了语义相似度分析，以期更好的实现成语反向查询，实验表明，本文检索能较好的满足用户的各类检索需求，可以尝试用于辅助写作。

当然，本文也还尚有不足，有很大的改进空间。对于正向成语查询，本文研究主要是在《中国成语大辞典》基础上进行了初步扩展，未来可以继续优化、完善成语词条资源。对于反向成语查询（基于语义的成语查询），未来可尝试的工作还有很多，主要可以从以下几个方面进行：词表示（词向量）模型的优化与改进；知识库等外部可用资源的进一步挖掘；打分方法的改进。

词表示模型训练方面，本文直接用了google开源的工具Word2vec，只是经过CBOW、Skip-gram等模型的理论学习和调研，最后选择了使用skip-gram模型来训练。然而目前各种深度神经网络技术已经很成熟，并应用在了自然语言处理的多项任务中，取得了各领域的state-of-the-art。比如说大家熟悉的深度神经网络（Deep Neural Networks）、长短时记忆模型的循环神经网络 ( Long-Short Term Memory Recurrent Neural Network, LSTM RNN) 以及注意力模型（Attention Model）等，未来可以尝试将这些模型用于词嵌入模型的训练，对比选择最好的模型，或是综合多种模型，以完整、全面的表示各个词。

研究资源方面，本文是从281.8G（约37亿字）微博、博客语料中筛选包含成语的子集来训练词向量模型的，后期可以丰富训练语料。在构建口语⬄成语资源库的时候，本文是基于《中国成语大辞典》进行补充得到的词条资源，但是仍旧有很多成语没能包含其中，后期可以借助百度成语以及其他词典资源继续完善。扩展口语表达的时候本文只用到了词向量计算和《同义词词林》 ，后期可以考虑利用《知网》[[8]](#footnote-8) 进行进一步扩展，对于《同义词词林》知识库也可以继续挖掘其可利用价值，比如本文只考虑了其中的相同词，还没有考虑近义词信息。

此外，打分排序阶段，本文使用的句子语义相似度计算方法只是基于关系向量模型进行了一些简单的改进，方法比较简单，需要更进一步地提出更优的方法，打分阶段也只是用到了查询query、成语释义、成语词条的语义信息，没有考虑备选成语在语料库中的频次因素，后续工作可以把它加入到打分计算。对于成语情感极性方面，本文只是借助微博语料的情感符号进行了简单的判断，目前情感计算方面的研究很多，SVM[36]、TD-IDF等也陆续被用于情感计算并取得了不错的效果，未来成语以及查询query极性判断方面，亦可以尝试使用深度学习的方法。

已发表文章

1. **Qu Qin**, Xun Endong, Yu Dong. Research and Experiment on Semantic-based Retrieval Methods of Idioms [M]//Chinese Lexical Semantic Workshop. International Journal of Knowledge and Language Processing Publishing, 2016: 15-23.

1. 百度百科：“成语”词条 [↑](#footnote-ref-1)
2. CSDN：Word2Vec的一些理解 [↑](#footnote-ref-2)
3. http://hanlp.linrunsoft.com/ [↑](#footnote-ref-3)
4. 百度百科：“停用词”词条 [↑](#footnote-ref-4)
5. http://more.datatang.com/data/39277 [↑](#footnote-ref-5)
6. http://more.datatang.com/data/44317 [↑](#footnote-ref-6)
7. http://more.datatang.com/data/44114 [↑](#footnote-ref-7)
8. http://www.keenage.com/html/c\_index.html [↑](#footnote-ref-8)