**SWordMap测试报告**

测试人：胡望胜

报告日期：2016-12-2

# 目的

编写该测试总结报告主要有以下几个目的：

1. 通过对测试结果的分析，得到对SWordMap有效性的评估。
2. 分析SWordMap存在的缺陷，为未来的工作展开提供建议。

# SWordMap简介

**研究背景：**现如今，随着软件规模的不断扩大，代码行数和参与人数的急剧增长，对软件的开发及维护变得相当困难。通常情况下，一个程序员需要依赖本地代码搜索来帮助快速定位到相关代码片段，大大加速软件开发及维护的工作。同时，随着越来越多的开源软件被开发，网络上已经有了海量开源代码，涵盖了方方面面，为帮助程序员理解、学习和重用代码提供了可能。为了在如此庞大的代码库中快速准确地找到相关代码，一个可靠的、自动化的开源代码搜索引擎是关键。

现有的代码搜索工具大部分是基于关键字文本匹配的搜索方法，面临与传统信息检索一样的问题，即用户查询关键字与代码文本用词不匹配。为提高搜索结果的精确度和召回率，需要对用户查询做语义相关词扩展。然而由于软件工程领域的单词语义与自然语言存在很大差异，代码搜索无法使用自然语言的语义相关词做查询扩展，需要软件工程领域的语义相关词表。目前已有的软件工程领域语义相关词挖掘研究如SWordNet，SEWordSim等，大多采用简单的文本相似度检测方法或基于词汇同现的统计方法，具有较大的局限性，对代码搜索精度的提升有限。

**技术特点：**为了能从单词的深层语义着手挖掘具有普适性的语义相关词，提高挖掘语义相关词的精确度，我们设计并实现了一种基于Word Embedding的软件工程领域语义相关词挖掘方法SWordMap。为研究SWordMap在代码搜索上的具体应用，我们设计并实现了针对代码搜索的查询扩展模型，以SWordMap挖掘语义相关词作为扩展词。

SWordMap的技术特点在于采用了自然语言领域的神经网络语言模型对软件工程领域文档进行训练，所得到的词向量能够表征单词深层语义，根据余弦距离计算所得的词向量距离能从语义层面表征两个单词的相关度。同时，分别针对本地代码搜索及开源代码搜索设计的查询扩展模型考虑了代码片段的特性及本地代码搜索与开源代码搜索的不同特点，与传统的直接将代码片段当作纯文本考虑、且不做查询扩展的代码搜索引擎相比能更准确地返回查询相关结果。

针对代码搜索的查询扩展模型是基于搜索引擎Elasticsearch实现的。Elasticsearch是一个分布式可扩展的实时搜索分析引擎，能够快速地处理大量数据的存储、搜索及分析，因此我们实现的查询扩展模型在搜索性能上也有一定保证，具有实用价值。

# SWordMap测试

SWordMap共设计了4个实验对其有效性进行验证评估。4个实验分别是：（1）SWordMap挖掘语义相关词表的精确度；（2）SWordMap对关注定位任务效率的提升；（3）SWordMap对本地代码搜索精度的提升；（4）SWordMap对开源代码搜索精度的提升。对这4个实验我们使用了不同的测试环境，测试数据，测试方法及测试标准。以下分别对这4个实验进行说明。

### 语义相关词表的精确度

该实验主要目标为评估SWordMap挖掘语义相关词表的精确度。这直接影响所得语义相关词表的实际应用价值。

##### 实验设定

1. 实验数据

从所得语义相关词表的19332个单词中随机选取100个单词，共1872个不重复的语义相关词对作为样本数据。

1. 实验方法

对样本数据中语义相关词对的正确性进行人工验证。为减少主观性的影响，验证由3名软件工程方向研究生完成，且采取较严格的评判标准，即三个验证者必须同时标记某个语义相关词对正确，才将该词对标记为正确。

在确认正确语义相关词对之后人工检查这些正确结果是否出现在Merriam-Webster英语词典或WordNet的语义相关词库中。选择Merriam-Webster英语词典是因为其权威性。

1. 实验评估标准

本实验采用评估标准为精确度（precision），即样本中正确语义相关词对数量占样本总数的比重。由于一个单词的语义相关词一共有多少难以统计，因此召回率无法衡量。

##### 实验结果

实验结果如表3-1所示。从表中数据可以看出样本的精确度为74.7%，说明SWordMap挖掘语义相关词词表的精确度平均在74.7%左右。同时，在所有正确语义相关词对中，不在Merriam-Webster英语词典或WordNet中的占75.3%，验证了单词语义在软件工程领域与自然语言之间的差异性。

表3-1 语义相关词正确性验证结果

Table 3-1 Experiment results of correctness verification of semantically related words

|  |  |
| --- | --- |
| 语义相关词对总数 | 1740 |
| 正确语义相关词对 | 1352 |
| 精确度（%） | 77.7 |
| 不在英语词典或WordNet中的正确结果数量 | 1011 |
| 不在英语词典或WordNet中的正确结果比例（%） | 74.8 |

### 关注定位

该实验主要目的为评估SWordMap挖掘语义相关词表是否能有效提升关注定位（concern location）任务效率。关注定位是指用户对软件中某一个功能或模块感兴趣，需要通过用户提供的关键字查找到软件中所有与该功能或模块相关的代码。

##### 实验设定

1. 实验环境

本实验采用的实验环境配置如表3-2所示。

表5-2 关注定位实验环境配置

Table 5-2 Experiment environment of concern location

|  |  |
| --- | --- |
| 配置 | 参数 |
| CPU | 2.4 GHz Intel Core i7 |
| 内存 | 8 GB 1600 MHz DDR3 |
| 操作系统 | Windows 8.1 |

1. 实验数据

关注定位的对象为表3-3所示的4个java软件。测试数据集如表3-4所示，为与这4个java软件相关的8个关注定位任务及结果。其中关注定位搜索的最小单位为代码中的函数。

表3-3 测试软件

Table 3-3 Evaluated java softwares

|  |  |  |  |
| --- | --- | --- | --- |
| 软件名称 | 软件描述 | 代码行数（LOC） | 函数个数 |
| iReport | 报表可视化设计器 | 74506 | 7587 |
| javaHMO | 多媒体服务器 | 25988 | 1787 |
| jBidWatcher | eBay拍卖监控软件 | 23502 | 1918 |
| jajuk | 音乐管理播放软件 | 20679 | 2132 |

1. 实验方法

搜索仅考虑函数的函数签名（函数名及其所属类名），搜索采取正则表达式匹配的搜索策略。具体来说，假设用户查询是“gather file”，那么原始查询的正则表达式是“.\*gather.\*file.\*”以及“.\*file.\*gather.\*”。在进行查询扩展时，假设collect是gather的一个语义相关词，那么查询扩展为“.\*gather.\*file.\*”，“.\*collect.\*file.\*”，“.\*file.\*gather.\*”以及“.\*file.\*collect.\*”。

1. 实验评估标准

本实验采用精确度（precision）及召回率（recall）作为评估标准。同时为了

表3-4 关注定位测试数据

Table 3-4 Concern location tasks

|  |  |
| --- | --- |
| 软件名称 | 关注定位任务 |
| iReport | add textfield |
| compile report |
| javaHMO | gather music files |
| load movie listing |
| jBidWatcher | add auction |
| save auction |
| set snipe price |
| jajuk | play song |

更直观地判断实验结果的好坏，本文还采用了F-measure评价指标。精确度即代码搜索返回结果中正确结果数量占所有返回结果总数的比重。召回率即代码搜索返回结果中正确结果数量占所有正确结果总数的比重。F-measure评价指标综合考虑了精确度及召回率，是二者的调和平均。F-measure的计算公式如下：

##### 实验结果

表3-5 关注定位实验结果

Table 3-5 Experiment results of concern location

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 关注定位任务 | 精确度（%） | | | 召回率（%） | | | F-measure | | |
| SWN | SWN-T | **SWM** | SWN | SWN-T | **SWM** | SWN | SWN-T | **SWM** |
| add textfield | 14.30 | 5.30 | 0 | 40 | 60 | 0 | 0.211 | 0.097 | 0 |
| compile report | 4 | 0.17 | 6.52 | 25 | 87.50 | 37.50 | 0.069 | 0.003 | 0.111 |
| gather music files | 28.60 | 0.50 | 28.57 | 50 | 75 | 50 | 0.364 | 0.010 | 0.364 |
| load movie listing | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add auction | 3.70 | 0.94 | 9.41 | 100 | 100 | 72.73 | 0.071 | 0.019 | 0.167 |
| save auction | 1.61 | 0.66 | 7.41 | 33.30 | 77.80 | 66.67 | 0.031 | 0.013 | 0.133 |
| set snipe price | 0 | 0 | 2.60 | 0 | 0 | 16.67 | 0 | 0 | 0.045 |
| play song | 0 | 0 | 11.54 | 0 | 0 | 75 | 0 | 0 | 0.200 |
| 平均 | 6.53 | 0.95 | **8.26** | 31.04 | 50.04 | **39.82** | 0.093 | 0.018 | **0.127** |

实验结果如表3-5所示。其中SWN表示SWordNet，SWN-T表示Transitive SWordNet，SWM表示SWordMap。

从表中列F-measure的数据可以看出，在8个搜索任务中有5个SWordMap的表现优于SWordNet，有6个优于Transitive SWordNet。根据F-measure指标，SWordMap相比SwordNet及Transitive SWordNet分别提升了36.6%及600%。

从表中数据可以看出，SWordMap可以识别SWordNet不能识别的语义相关词从而提高代码搜索召回率，如查询“play song”中，SWordMap成功将查询中的song和相关结果中的playlist识别成一对语义相关词而SWordNet没有。这是由于SWordNet采用简单的文本相似度比较挖掘语义相关词，无法真正理解单词的深层语义。而且SWordNet挖掘对象仅限于软件代码及注释，无法挖掘软件文档中的语义相关词。SWordMap还能有效提升代码搜索精确度，8个搜索任务中有5个SWordMap精确度优于SWordNet，有6个优于Transitive SWordNet。

SWordMap仍存在缺陷。在查询“add textfield”中，SWordMap未能找到任何相关结果。这是因为该查询中相关结果包含关键字为drop，SWordMap未能识别add和drop为语义相关。add和drop这对语义相关词是软件iReport特有的，不具备普适性，因此SWordMap未能识别。在查询“load movie listing”中，SWordMap与SWordNet表现一样，未能找到任何相关代码。这是因为该查询中相关结果包含关键字为reload和container，SWordMap成功识别了load和reload为语义相关，未能识别listing和container。这说明SWordMap仅使用Stack Overflow的文档作为训练语料库是不充分的。SWordMap的平均召回率要低于Transitive SWordNet。注意虽然Transitive SWordNet的召回率较高，但它的精确度极低。实验中发现，使用Transitive SWordNet做查询扩展会导致代码搜索返回结果数量剧增（个别查询返回结果数量接近软件中所有函数），极大影响搜索的实用价值。

### 本地代码搜索

该实验主要目的为评估SWordMap挖掘语义相关词表是否能有效提升本地代码搜索精度。许多软件任务均从本地代码搜索开始，其返回结果给程序员提供了一个起点。

##### 实验设定

1. 实验环境

本实验采用的实验环境配置同3.2.1节，如表3-2所示。

1. 实验数据

本地代码搜索的对象与关注定位实验相同，为表3-3中的4个java软件。测试数据集的构建由3名软件工程方向研究生共同完成。测试数据中的查询任务由其中1名研究生提出，该研究生事先没有对测试软件的任何了解，在阅读过测试软件的相关文档后提出了这些与软件具体功能相关的问题。为减少主观性的影响，其他2名研究生对这些查询的客观性进行验证，检查它们是否带有对某项技术的偏好。最终选取了共25个查询。之后3名研究生对这些查询的相关结果的真实数据（ground truth）集进行构建。对于一个查询，这3名研究生首先分别阅读软件源码，查找它的相关结果，然后对3人查找结果进行汇总：首先取3人查找结果的交集，对3人查找结果并集中的其他结果由3人一起进行验证，只有在3人均同意该结果为真的时候才将其加入真实数据集。测试数据集如表3-6所示。

1. 实验方法

采用针对本地代码搜索的查询扩展模型，分别使用SWordNet及SWordMap

表3-6 本地代码搜索测试数据

Table 3-6 Local code search tasks

|  |  |
| --- | --- |
| 软件名称 | 查询任务 |
| iReport | initialize a chart dialog |
| get the chosen text |
| get db connection credentials |
| export to excel |
| set attributes of a chart |
| copy a dataset |
| javaHMO | load audio data |
| collect all music files |
| get the category of a music |
| get the thumbnail of a movie |
| set the singer of a song |
| extract network domain information |
| configure server ip address |
| find image files |
| jBidWatcher | delete an auction |
| get the best bid price |
| update an auction |
| buy an item |
| login to ebay |
| translate between currency |
| jajuk | get playable songs |
| add a playlist |
| remove a song from playlist |
| get the author of a track |
| silence the player |

挖掘的语义相关词进行代码搜索。同时设立不做查询扩展的对照组。具体来说，假设用户查询为“export to excel”，则去除停词后原始查询为“export AND excel”。在进行查询扩展时，假设export的语义相关词是exporter，excel的语义相关词是csv，那么查询扩展为“(export OR exporter) AND (excel OR csv)”。

1. 实验评估标准

与3.2.1节相同，采用精确度及召回率作为评估标准。

##### 实验结果

实验结果如表3-7所示。其中Initial表示不做查询扩展的对照组，SWN表

示SWordNet，SWN-T表示Transitive SWordNet（考虑了语义相关词对传递性的SWordNet），SWM表示本文工作SWordMap。

从表中数据可以看出，采用SWordMap查询扩展能有效提升本地代码搜索精度。SWordMap的精确度为37%，覆盖率为77.22%，相比原始查询分别提升了147.8%及683.2%。SWordMap相比SWordNet和Transitive SWordNet也有显著提升。

注意到对许多查询SWordNet找不到任何相关结果而SWordMap可以。这是由于SWordNet仅能挖掘出现在当前搜索的软件中的语义相关词，若一个语义相关词未在当前搜索的软件中出现，则SWordNet无能为力。如软件iReport的查询“export to excel”中，excel与csv为一对语义相关词，由于excel没有出现在软件iReport中，SWordNet未能成功识别。SWordMap能识别的语义相关词范围取决于训练语料库，只要训练语料库范围足够广泛就不会有这个问题。同时在所有查询上SWordNet的精确度均低于原始查询，说明SWordNet挖掘的语义相关词的精确度较低，简单的文本相似度比较导致了较多误报，虽然在召回率上有提升，但精确度的大幅下降导致搜索的实用价值极低。相比SWordNet，SWordMap在具有高召回率的同时还有着较高的精确度，能够给本地代码搜索的用户带来可观的帮助。

SWordMap仍有其局限性。在查询“buy an item”中，SWordMap未能识别软件jBidWathcer特有的语义相关词对item和auction。在查询“login to ebay”中，SwordMap未能识别语义相关词对login和sign，再次说明了SWordMap训练语料库不够充分。在查询“get playable songs”中，一个有趣的事实是软件jajuk的代码中错误地将playable拼写成了playeable，导致SWordMap及SWordNet均未能找到相关结果。这也证实了拼写错误在代码中的确存在，因此识别一个单词的拼写错误对代码搜索也是有帮助的。在查询“collect all music files”中，相关结果中包含的关键字为gather和directory，SWordMap成功识别了语义相关词对collect和gather以及files和directory，但是由于相关结果不包含music或其语义相关词，搜索未能返回任何相关结果。这实际上是检索模型的不足。

表3-7 本地代码搜索实验结果

Table 3-7 Experiment results of local code search

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 查询任务 | 精确度（%） | | | | 召回率（%） | | | |
| Initial | SWN | SWN-T | **SWM** | Initial | SWN | SWN-T | **SWM** |
| initialize a chart dialog | 0 | 0 | 0 | 54.55 | 0 | 0 | 0 | 100 |
| get the chosen text | 0 | 0 | 0 | 31.82 | 0 | 0 | 0 | 100 |
| get db connection credentials | 0 | 0 | 0 | 40 | 0 | 0 | 0 | 100 |
| export to excel | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 100 |
| set attributes of a chart | 0 | 0 | 0 | 14.81 | 0 | 0 | 0 | 57.14 |
| copy a dataset | 16.67 | 0.76 | 0.13 | 9.46 | 14.29 | 14.29 | 100 | 100 |
| load audio data | 100 | 25 | 0.26 | 66.67 | 20 | 60 | 60 | 40 |
| collect all music files | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| get the category of a music | 0 | 0 | 0 | 9.09 | 0 | 0 | 0 | 100 |
| get the thumbnail of a movie | 0 | 4.76 | 0.20 | 7.69 | 0 | 100 | 100 | 100 |
| set the singer of a song | 0 | 0 | 0 | 8.70 | 0 | 0 | 0 | 100 |
| extract network domain information | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 100 |
| configure server ip address | 0 | 0 | 0 | 16.67 | 0 | 0 | 0 | 100 |
| find image files | 0 | 2.82 | 0.17 | 25 | 0 | 100 | 100 | 100 |
| delete an auction | 66.67 | 0.61 | 0.38 | 10 | 33.33 | 83.33 | 100 | 100 |
| get the best bid price | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 100 |
| update an auction | 50 | 1.31 | 0.66 | 17.74 | 45.45 | 100 | 100 | 100 |
| buy an item | 0 | 0.56 | 0.26 | 0 | 0 | 75 | 100 | 0 |
| login to ebay | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| translate between currency | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 100 |
| get playable songs | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add a playlist | 40 | 10.17 | 2.60 | 4.17 | 33.33 | 100 | 100 | 33.33 |
| remove a song from playlist | 0 | 0 | 0 | 8.70 | 0 | 0 | 0 | 100 |
| get the author of a track | 100 | 0.51 | 0.25 | 100 | 100 | 100 | 100 | 100 |
| silence the player | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 100 |
| 平均 | 14.93 | 1.86 | 0.20 | **37** | 9.86 | 29.30 | 34.40 | **77.22** |

### 开源代码搜索

该实验主要目的为评估SWordMap挖掘语义相关词表是否能有效提升开源代码搜索精度。开源代码搜索为程序员理解、学习和重用代码提供了帮助，让程序员可以快速学习新的API，借鉴前人工作来解决软件开发中遇到的问题。

##### 实验设定

1. 实验环境

本实验采用的实验环境为两台服务器，配置如表3-8所示。每台服务器上部署了两个Elasticsearch节点，一共四个节点。

表3-8 开源代码搜索实验环境配置

Table 3-8 Experiment environment of open-source code search

|  |  |
| --- | --- |
| 配置 | 参数 |
| CPU | 2.4 GHz Intel Core i7 |
| 内存 | 256 GB 1600 MHz DDR3 |
| 操作系统 | Windows Server 2012 |

1. 实验数据

开源代码搜索的对象为开源社区Github的java项目。Github作为全球流行的开源社区，其上有着海量的具有高关注度及高可靠性的开源项目。Github上的项目具有收藏数（star），一般来说收藏数越大说明该项目越受关注，代码质量越高。本文从Github上面下载了收藏数大于900的共1023个java项目，共约20.6GB，包含457265个java代码文件，3547798个函数，其中经过过滤后被索引到Elasticsearch集群中的函数共1304201个。

测试采用的查询为表3-9中的34条查询。其中，第1-4条取自前人研究工作，第5-22条取自IT技术问答网站Stack Overflow，第23-34条取自Bing搜索日志。

1. 实验方法

采用针对开源代码搜索的查询扩展模型，使用SWordMap挖掘的语义相关词进行代码搜索。同时设立不做查询扩展的对照组。具体来说，假设用户查询为“export to excel”，则去除停词后原始查询为“remove cookie”。在进行查询扩展时，假设remove的语义相关词是delete，cookie的语义相关词是session，那么查询扩展为“(remove OR delete) AND (cookie OR session)”。

1. 实验评估标准

本实验采用Precision@k及MRR（Mean Reciprocal Rank）指标作为评估标准。Precision@k用来衡量返回结果的前k个中相关结果的数量，是所有查询的前k个返回结果中相关结果所占比重的平均。Precision@k的计算公式如下：

其中表示第i个查询的前k个返回结果中相关结果数量。Precision@k越大说明代码搜索精度越高。本文选取的k值为1，5，10及20。多数情况下用户不会关注排在第20之后的结果。MRR用来衡量返回结果中出现的第一个相关结果的排序，是所有查询的返回结果中第一个相关结果排序倒数的平

表3-9 开源代码搜索测试数据

Table 3-9 open-source code search queries

|  |  |
| --- | --- |
| 来源 | 查询 |
| 前人研究 | copy paste data from clipboard |
| open url in html browser |
| track mouse hover |
| highlight text range in editor |
| Stack Overflow | convert utc time to local time |
| converting String to DateTime |
| get current date and time |
| get file name without extension |
| how can I decode HTML characters |
| how can I download HTML source |
| how do I round a decimal value to 2 decimal places |
| how to change RGB color to HSV |
| how to convert an IPv4 address into a integer |
| how to delete all files and folders in a directory |
| how to execute a sql select |
| how to generate random int number |
| how to get Color from Hexadecimal color code |
| how to get temporary folder for current user |
| if a folder does not exist create it |
| ping a hostname on the network |
| Process.start: how to get the output |
| sending email through Gmail |
| Bing搜索日志 | append string to file |
| calculate md5 checksum |
| how to deserialize XML document |
| how to get mac address |
| how to play a sound |
| how to save image in png format |
| read file line by line |
| remove cookie |
| verify folder exists |
| how to reverse a string |
| quick sort |
| how to split string into words |

均。MRR的计算公式如下：

其中表示第i个查询的返回结果中第一个相关结果的排序。MRR越大说明代码搜索精度越高。

返回结果的相关性判断由3名软件工程方向研究生共同完成。对一个返回结果，只有当3人均认为其与查询相关时才被标记为相关。

##### 实验结果

实验结果具体数据如表3-10所示。评估标准实验结果如表3-11所示。其中Initial表示不做查询扩展的对照组，SWordMap表示做查询扩展的实验组。

从表3-10中数据可以看出，在所有查询中的18条查询上SWordMap表现优于原始查询，7条查询上SWordMap表现与原始查询相当，9条查询上SWordMap表现比原始查询差。对其中一些查询，SWordMap能较大提升返回结果精度。如查询“how can I decode HTML characters”，SWordMap能识别decode的语义相关词encode、parse等等，html的语义相关词webpage、css、tag等等，以及characters的语义相关词chars、symbols等等，因此能够大幅度提升相关结果的排序。又如查询“how to convert an IPv4 address into a integer”，SWordMap能识别ipv4的语义相关词ipv6、ip等等，以及address的语义相关词addr、hostname等等，因此前20个返回结果中相关结果数量有较大提升。对其中一些查询，SWordMap仅有较小提升或没有提升。如查询“how to reverse a string”、“quick sort”等等，这是因为原始查询本身即能较好匹配到大量相关结果。而对余下查询，SWordMap表现要差于原始查询。如查询“how to get mac address”，由于mac本身具有两种不同语义：（1）苹果电脑的操作系统；（2）网卡物理地址，SWordMap识别的mac的语义为第一种，所识别的语义相关词为osx、ubuntu、windows等等，导致搜索结果精度大大下降。这说明两个单词的相关性是受上下文限制的，不同上下文中两个单词的语义相关性也不同。

从表3-11中数据可以看出，总的来说SWordMap相对原始查询在代码搜索精度上有一定提升。其中，Precision@1提升了8.64%，Precision@5提升了5.23%，Precision@10提升了5.13%，Precision@20提升了8.20%，MRR提升了5.04%。注意虽然结果有提升，但提升并不显著，相比本地代码搜索实验结果的提升是很小的。这是由于在本地代码搜索中，搜索对象数量很少，仅是一个软件中的所有代码，因此在查询扩展时部分错误的语义相关词对搜索结果不会产生很大影响（大部分错误的语义相关词不会在软件中出现）。而在开源代码搜索中，搜索对象数量巨大，在查询扩展时错误的语义相关词也会在许多代码片段中出现，因而搜索结果会受到较大影响。同时由于单词的语义相关词受上下文限制，即使是对

表3-10 开源代码搜索实验结果

Table 3-10 Experiment results of open-source code search

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 查询任务 | 前k个返回结果中相关结果数量 | | | | | | | |
| Initial | | | | SWordMap | | | |
| k=1 | k=5 | k=  10 | k=  20 | k=1 | k=5 | k=  10 | k=  20 |
| copy paste data from clipboard | 1 | 5 | 9 | 15 | 1 | 5 | 9 | 17 |
| open url in html browser | 1 | 2 | 6 | 11 | 1 | 4 | 8 | 14 |
| track mouse hover | 1 | 4 | 7 | 12 | 1 | 2 | 5 | 9 |
| highlight text range in editor | 1 | 4 | 7 | 13 | 1 | 2 | 3 | 7 |
| convert utc time to local time | 0 | 3 | 6 | 11 | 1 | 4 | 6 | 13 |
| converting String to DateTime | 1 | 4 | 4 | 8 | 1 | 4 | 5 | 9 |
| get current date and time | 1 | 3 | 6 | 13 | 1 | 5 | 8 | 15 |
| get file name without extension | 1 | 5 | 9 | 13 | 1 | 5 | 9 | 15 |
| how can I decode HTML characters | 1 | 4 | 4 | 4 | 1 | 5 | 9 | 14 |
| how can I download HTML source | 1 | 2 | 3 | 7 | 0 | 1 | 1 | 5 |
| how do I round a decimal value to 2 decimal places | 0 | 2 | 4 | 7 | 0 | 3 | 7 | 14 |
| how to change RGB color to HSV | 1 | 4 | 5 | 8 | 1 | 2 | 6 | 7 |
| how to convert an IPv4 address into a integer | 0 | 2 | 3 | 5 | 0 | 4 | 7 | 15 |
| how to delete all files and folders in a directory | 1 | 5 | 9 | 10 | 1 | 5 | 9 | 12 |
| how to execute a sql select | 0 | 2 | 5 | 7 | 0 | 4 | 8 | 10 |
| how to generate random int number | 1 | 4 | 8 | 17 | 1 | 4 | 8 | 17 |
| how to get Color from Hexadecimal color code | 0 | 2 | 4 | 7 | 1 | 2 | 3 | 7 |
| how to get temporary folder for current user | 0 | 2 | 3 | 5 | 1 | 2 | 3 | 6 |
| if a folder does not exist create it | 1 | 4 | 7 | 15 | 1 | 4 | 7 | 15 |
| ping a hostname on the network | 1 | 4 | 8 | 13 | 1 | 4 | 8 | 15 |
| Process.start: how to get the output | 0 | 3 | 8 | 9 | 0 | 3 | 7 | 9 |
| sending email through Gmail | 1 | 3 | 5 | 6 | 1 | 4 | 7 | 16 |
| append string to file | 1 | 5 | 6 | 13 | 1 | 5 | 6 | 11 |
| calculate md5 checksum | 0 | 2 | 5 | 9 | 0 | 3 | 5 | 7 |
| how to deserialize XML document | 1 | 4 | 9 | 12 | 0 | 3 | 8 | 10 |
| how to get mac address | 1 | 2 | 3 | 11 | 1 | 2 | 2 | 3 |
| how to play a sound | 1 | 4 | 8 | 15 | 1 | 4 | 8 | 16 |
| how to save image in png format | 1 | 5 | 9 | 18 | 1 | 5 | 8 | 16 |
| read file line by line | 1 | 3 | 5 | 10 | 1 | 4 | 5 | 10 |
| remove cookie | 0 | 2 | 3 | 5 | 0 | 2 | 3 | 5 |
| verify folder exists | 0 | 3 | 7 | 12 | 1 | 3 | 8 | 13 |
| how to reverse a string | 1 | 5 | 8 | 16 | 1 | 5 | 8 | 16 |
| quick sort | 1 | 5 | 10 | 18 | 1 | 5 | 10 | 18 |
| how to split string into words | 1 | 4 | 9 | 18 | 1 | 4 | 9 | 18 |

表5-11 开源代码搜索评估标准实验结果

Table 5-11 Experiment results of evaluation metrics of open-source code search

|  |  |  |
| --- | --- | --- |
|  | Initial | SWordMap |
| Precision@1 | 0.706 | 0.767 |
| Precision@5 | 0.688 | 0.724 |
| Precision@10 | 0.624 | 0.656 |
| Precision@20 | 0.549 | 0.594 |
| MRR | 0.814 | 0.855 |

的语义相关词也不一定会带来好的搜索结果。这意味着在做查询扩展时还需要考虑原始查询的上下文。

# 测试结果分析

针对SWordMap的4个实验：第一个实验评估SWordMap挖掘语义相关词表的精确度，实验结果表明SWordMap挖掘语义相关词表精确度较高。第二个实验评估SWordMap对关注定位任务效率的提升，实验结果表明SWordMap能有效提升关注定位任务效率。第三个实验评估SWordMap对本地代码搜索精度的提升，实验结果表明SWordMap能有效提升本地代码搜索精度。第四个实验评估SWordMap对开源代码搜索精度的提升，实验结果表明SWordMap对开源代码搜索精度有一定提升，但效果不显著。总的来说，SWordMap挖掘语义相关词具有一定应用价值，相比开源代码搜索更适用于本地代码搜索。

# 测试结论及建议

我们共设计了四个实验，从不同角度评估SWordMap的有效性。实验结果表明SWordMap挖掘语义相关词精确度较高，能有效提升关注定位任务效率及本地代码搜索精度，对开源代码搜索精度的提升有限。

根据SWordMap的实验结果，SWordMap目前还存在一些缺陷。针对这些缺陷SWordMap有着下述改进空间：

1. 扩充训练数据

关注定位及本地代码搜索的实验结果均表明SWordMap挖掘的语义相关词存在遗漏，说明SWordMap仅使用Stack Overflow的文档作为训练数据不够充分。在将来的工作中SWordMap需考虑更多的训练数据来源，如软件开发者手册、API文档、开源软件的代码及文档等等。

1. 改进训练模型

目前的CBOW训练模型仅能对具有句子格式的软件文档进行训练，对代码的训练结果较差。因此在将来的工作中可以考虑对CBOW模型进行改进，使得其能够对代码这种结构化文本进行训练。

1. 改进查询扩展模型

目前的查询扩展模型未考虑查询中的上下文，导致在开源代码搜索中的表现不佳。因此可以对查询扩展模型进行改进，在选取扩展词时考虑查询上下文，确保扩展词的确与查询语义上相关。一个简单的想法如直接计算查询中所有单词与词库中单词的向量距离，选取词库中与所有查询中单词距离的均值最近的单词作为扩展词。

同时目前的查询扩展模型仅考虑了代码文本信息，未考虑代码结构信息。在将来的工作中可以考虑将代码结构也作为一个代码特性进行搜索。

综上，SWordMap能够有效挖掘软件工程领域语义相关词，能有效提升关注定位任务效率及本地代码搜索精度，但是对开源代码搜索精度的提升有限。SWordMap仍存在提升空间，可以提高其对开源代码搜索的应用价值。