相似度计算实验

1. 实验基本内容

包管理工具是在使用 Linux 操作系统时的常用工具，通过包管理工具我们可以便捷地安装与卸载软件，而不需要像使用 Windows 系统时那样自行管理各种软件及其依赖关系。不同版本的 Linux 系统之间是有不同的包管理工具的，例如 Ubuntu 中的 apt，centOS 中的 yum，openEuler 的 DNF 等，他们背后的原理相似但所使用的软件源不同，互相之间有的软件有对应关系，有的则没有直接的对应软件。

本实验旨在让同学们了解包管理工具是如何帮助用户进行软件的下载，即包管理工具如何从镜像源下载压缩好的软件包并解压，熟悉 Linux 中软件包有哪些特征，并且探索不同的 Linux 发行版的软件包之间的映射关系。当然，包管理工具从软件源将软件包下载好并解压只是安装软件的第一步，后续的安装与维护还有很多步骤，有兴趣的同学可以自行探索。

在本实验中，需要同学们计算 CentOS（From OS）与 openEuler（To OS）之间软件包的映射关系，具体任务如下：

a.下载 CentOS 的所有软件包，CentOS 的版本为 7.8.2003

b.下载 openEuler 的所有软件包，openEuler 的版本为 22.03-LTS

c.将下载好的软件包解压并存储在合适的位置（可能需要预留较大的存储空间，400GB 以上）

d.确定自己的映射计算方案，例如，可以计算两个系统之间的软件包相似度，当相似度大于一个阈值时，就判定这一对软件包为对应关系，即可确定为映射，如果未达到则判定为无映射关系。

e.存储最终的映射计算结果

2. 实验准备

本节主要介绍实验运行的基本环境，并给出软件包下载的思路提示。软件包需要到可用的软件源进行下载，国内目前常用的 Linux 软件源主要有华为源、阿里源、清华源等，同学们可以根据自身喜好与下载速度自行选择一个下载。

2.1 实验运行环境

● openEuler 版本：24.03 LTS

● Python 版本：Python 3.9 及以上版本（为确保能够支持实验所需的依赖库，应使用较新的 Python 版本）。

2.2 软件包下载

1. 编写脚本下载 Centos 软件包：

在 openEuler 操作系统中，创建一个用于存放下载的 rpm 文件的目录

（在 centos 中，源码包的存储文件为 rpm 文件和 primary.xml 文件，二进制包的存储文件为 filelists.xml 文件）：

```
mkdir -p /c/pkgmapping/centos/7.8.2003/srpm/Packages
chmod 775 /c/pkgmapping/centos/7.8.2003/srpm/Packages
```

自行编写脚本 download_centos_7_src_rpm.sh 下载 Centos 7 的软件包。以下是一些提示和步骤：

提示 1：

定义基础目录 base_dir，镜像地址 source，CentOS 版本 version 和包类型数组 package_type.

```
base_dir="/c/pkgmapping/centos"
source="https://mirrors.huaweicloud.com/centos-vault"
version="7.8.2003"
packages_type=(updates os extras)
```

提示 2：

下载源 RPM 包，检查并创建 ${base_dir}/${version}/srpm/Packages 目录（base_dir,version 同提示一中的代码块），设置权限为 775，遍历 packages_type 数组，构建仓库地址并使用 lftp 下载所有匹配 *.src.rpm 的文件

提示 3：

下载 RPM 元数据 XML 文件。检查并创建 ${base_dir}/${version}/rpmxml 目录，设置权限为 775。遍历 packages_type 数组，构建仓库地址并使用 lftp 下载所有匹配 *primary.xml.gz 的文件。使用 gzip -d 解压下载的 XML 文件。

提示 4：

下载二进制元数据 XML 文件。检查并创建 ${base_dir}/${version}/binaryxml 目录（base_dir,version 同提示一中的代码块），设置权限为 775。遍历 packages_type 数组，构建仓库地址并使用 lftp 下载所有匹配 *filelists.xml.gz 的文件。使用 gzip -d 解压下载的 XML 文件。

提示 5：

CentOS 的软件包位置与 openEuler 的软件包位置并不相同，openEuler 中的软件包数据请在下面的链接中下载：

source：mirrors.huaweicloud.com/openeuler，version：openEuler-22.03-LTS

package_type:aarch64,x86_64

rpm 包（所有路径数据之和）	${source}/${version}/source/Packages/ ${source}/${version}/EPOL/main/source/Packages/ ${source}/${version}/EPOL/update/main/source/Packages/ ${source}/${version}/update/source/Packages/
rpm 元数据（所有路径数据之和）	${source}/${version}/source/repodata/ ${source}/${version}/EPOL/main/source/repodata/ ${source}/${version}/EPOL/update/main/source/repodata/ ${source}/${version}/update/source/repodata/
二进制元数据（所有路径数据之和）	${source}/${version}/everything/${package_type}/repodata/ ${source}/${version}/EPOL/main/${package_type}/repodata/ ${source}/${version}/EPOL/update/main/${package_type}/repodata/ ${source}/${version}/update/${package_type}/repodata/

确保脚本能够正确执行上述提示步骤，下载并解压相应的文件。

2. 解压源码包、二进制包

由于 OS 从镜像源中下载的源码包、二进制包是压缩文件，因此需要编写脚本来解压

提示 1：使用 find 命令查找所有下载的压缩文件。可以使用 find 命令检查路径中所有文件的路径，结合 while 循环进行处理。

提示 2：利用 basename 命令获取文件名，并可通过 rpm2cpio 命令把 RPM 包转换为 CPIO 归档文件，再使用 cpio 命令将 CPIO 归档文件中的内容提取到指定的目标目录。

提示 3：编写一个 if 分支，根据文件扩展名（如.gz、.bz2、.xz）选择适当的解压参数。

提示 4：使用 tar 命令解压文件，并将解压后的文件保存到指定的目录。

3. 提取特征

从下载的源码包和二进制包中提取出后续映射所需要的特征信息，并将这些特征信息存储在对应的 json 文件中。

源码包理应包含的特征信息：

```
{
"name": "",
"version": "",
"summary": "",
"description": "",
"url": "",
"requires": [],
"providers": [],
"binaryList": [],
"buildRequires": [],
"source0": "",
"macro_names": [],
"email_names": [],
"class_names": [],
"path_names": [],
"url_names": []
}
```

二进制包理应包含的特征信息

```
{ "name": "", "arch": "", "filelist": [] }
```

对于**源码包**，尝试编写 python 脚本 extract_rpm.py 来完成目标

```
python feature_extract/extract_rpm.py
```

希望可以通过上面的命令来生成“XXX_rpm.json”文件。

之后尝试编写 python 脚本 extract_spec.py 来完成目标

```
python feature_extract/extract_spec.py
```

希望可以通过上面的命令来生成“XXX_spec.json”文件。

当然，还需要编写一个合并特征的脚本 merge_json.py，预期的效果是通过命令

```
python utils/merge_json.py -rpm "XXX_rpm.json 的文件路径" -spec "XXX_spec.json 的文件路径" -merge "合并后的 json 的文件路径"
```

来将两个 json 文件整合在同一个 json 文件中，实现对源码包特征的提取。

对于**二进制包**，编写 extract_bin.py，预期是通过命令

```
python feature_extract/extract_bin.py
```

来生成“XXX_binary.json”文件

当然，对于上述的所有命令（除合并 json 的程序），同学们也可以将 os 的类型作为命令行参数传入，对于不同系统，分别采用如下的调用方式实现程序的运行：

```
python feature_extract/extract_bin.py -o "OS 的名称"
```

3. 预期输出

请在实验过程中尽量保留每一步截图

检查：

1. 设置参数（使用深度学习进行计算的话）：

    1.1 在 config/constant.py 中对上述特征信息的值进行特征权重赋予。给出用到的特征信息以及其对应的权重，并简要解释为何这么设置

    1.2 在 config/constant.py 中设置相似度阈值，一般来说为 0

    1.3 设置特征计算函数：给出用到的特征相似度计算函数并对比其他的相似度计算函数。解释为何最终使用该特征计算函数。

2. 映射结果

    执行代码进行映射，映射结果存储在/data/mapping 中，命名方式为

    {源 OS 名称}@{源 OS 版本号}-to-{目标 OS 名称}@{目标 OS 版本号}

    例如：archlinux@1-to-openeuler@20.03.txt

    需要输出的内容有以下几列：

```
idx | from_os | from_pkg | to_os | to_pkg | similarity | valid_pkg | right |
```

- **idx**: 序号，标识每一行数据的唯一编号。
- **from_os**: 来源操作系统，表示该软件包属于哪个操作系统及其版本，例如 archlinux@1 表示 Arch Linux 操作系统的版本 1。
- **from_pkg**: 来源软件包，指示在来源操作系统中要映射的软件包名称，例如 aalib。
- **to_os**: 目标操作系统，表示映射的目标操作系统及其版本，例如 openeuler@20.03 表示 OpenEuler 操作系统的版本 20.03。
- **to_pkg**: 目标软件包，表示在目标操作系统中对应的软件包名称，若为空，则表示目标操作系统中未找到相应的软件包。
- **similarity**: 相似度，表示来源软件包与目标软件包之间的相似程度，通常以百分比表示。
- **valid_pkg**: 有效软件包，表示该条映射是否有有效的目标软件包名称。若目标软件包有效，列出软件包名称；若无效，可能为空或标记为无效。
- **right**: 判断结果，表示该条映射是否正确。若映射是有效的且正确，显示为 True，否则显示为 False。

例如：

idx	from os	from pkg	to os	to pkg	similarity	valid pkg	right?
0	archlinux@1	aalib	openeuler@20.03		29.40%	aalib	False

idx = 0 的一行，表示从 archlinux@1 操作系统中的 aalib 软件包映射到 openeuler@20.03 操作系统中，但目标软件包名称为空，相似度为 29.40%，且该映射被判断为错误 (False)。

成功下载源码包、二进制包：

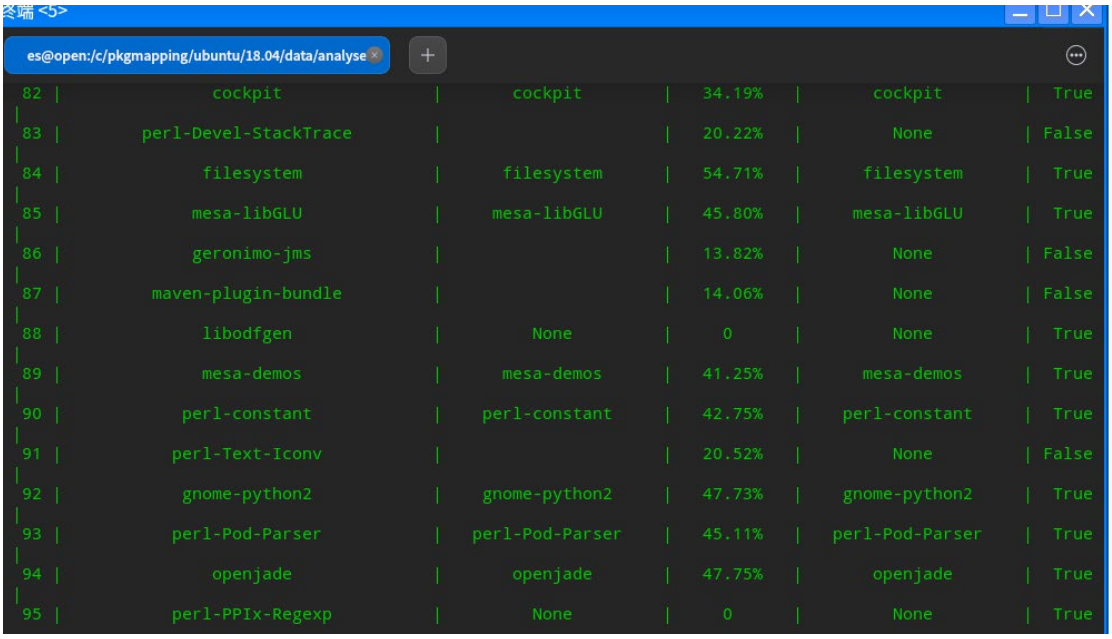
```
[root@open 18.04]# bash get_package/download_centos_7_src_rpm.sh
cd 成功，当前目录=/centos-vault/7.8.2003/updates/Source/SPackages
8268970254 bytes transferred in 864 seconds (9.12 MiB/s)
Total 106 files transferred
cd 成功，当前目录=/centos-vault/7.8.2003/os/Source/SPackages
10400227683 bytes transferred in 1739 seconds (5.70 MiB/s)
Total 2748 files transferred
cd 成功，当前目录=/centos-vault/7.8.2003/extras/Source/SPackages
243815186 bytes transferred in 44 seconds (5.26 MiB/s)
Total 61 files transferred
cd 成功，当前目录=/centos-vault/7.8.2003/updates/Source/repodata
33212 bytes transferred
cd 成功，当前目录=/centos-vault/7.8.2003/os/Source/repodata
751356 bytes transferred
cd 成功，当前目录=/centos-vault/7.8.2003/extras/Source/repodata
12371 bytes transferred in 4 seconds (3.3 KiB/s)
cd 成功，当前目录=/centos-vault/7.8.2003/updates/x86_64/repodata
2598045 bytes transferred
cd 成功，当前目录=/centos-vault/7.8.2003/os/x86_64/repodata
7469883 bytes transferred
cd 成功，当前目录=/centos-vault/7.8.2003/extras/x86_64/repodata
185404 bytes transferred
[root@open 18.04]#
```

映射结果检验：

①无效的

```
| 105 | centos@7.8.2003 |             tbb             | openeuler@20.03 |                           |   26
.61%   |          tbb          | False  |
| 106 | centos@7.8.2003 |            totem           | openeuler@20.03 |                           |   23
.98%   |     totem-pl-parser     | False  |
| 107 | centos@7.8.2003 |          virt-manager         | openeuler@20.03 |                           |   24
.86%   |       virt-viewer       | False  |
| 108 | centos@7.8.2003 |             vte3            | openeuler@20.03 |           vte291          |   30
.57%   |       geocode-glib      | False  |
| 109 | centos@7.8.2003 |       xorg-sgml-doctools      | openeuler@20.03 |                           |   27
.66%   |          imake          | False  |
| 110 | centos@7.8.2003 |         xorg-x11-docs         | openeuler@20.03 |        xorg-x11-fonts       |   30
.06%   |          imake          | False  |
| 111 | centos@7.8.2003 |     xorg-x11-drv-keyboard     | openeuler@20.03 |     xorg-x11-drv-vmware     |   32
.50%   |          imake          | False  |
| 112 | centos@7.8.2003 |       xorg-x11-drv-mouse      | openeuler@20.03 |     xorg-x11-drv-vmware     |   33
.92%   |          imake          | False  |
| 113 | centos@7.8.2003 |     xorg-x11-drv-synaptics    | openeuler@20.03 |                           |   27
.59%   |          imake          | False  |
| 114 | centos@7.8.2003 |      xorg-x11-drv-vmmouse     | openeuler@20.03 |     xorg-x11-drv-vmware     |   34
.47%   |          imake          | False  |
| 115 | centos@7.8.2003 |       xorg-x11-drv-void       | openeuler@20.03 |     xorg-x11-drv-vmware     |   33
.49%   |          imake          | False  |
| 116 | centos@7.8.2003 |        yum-rhn-plugin         | openeuler@20.03 |                           |   24
.04%   |          rhnlib         | False  |
```

②随机抽取 100 条数据检验



3. 项目源码

   验收时应提供完整的、可运行的源码，并讲解代码思路。

4. checklist

   下载链接：https://box.nju.edu.cn/f/de3677d893f3434ea411/?dl=1

+-----+------------------------+------------------------+

| idx | from_pkg | to_pkg |

+-----+------------------------+------------------------+

| 1 | adobe-mappings-cmap | adobe-mappings-cmap-lang |

+-----+------------------------+------------------------+

| 2 | atkmm-doc | atkmm-help |

+-----+------------------------+------------------------+

| 3 | bpg-mrgvlovani-caps-fonts | bpg-mrgvlovani-fonts |

+-----+------------------------+------------------------+

| 4 | bpg-nino-medium-cond-fonts | bpg-nino-medium-fonts |

+-----+------------------------+------------------------+

| 5 | cairomm-doc | cairomm-help |

+-----+------------------------+------------------------+

| 6 | cockpit-system | cockpit |

+-----+------------------------+------------------------+

| 7 | cogl-doc | cogl-help |

```
+-----+------------------------+------------------------+
|     8 | colord-devel-docs               | colord-help                      |
+-----+------------------------+------------------------+
|     9 | colord-extra-profiles        | colord-devel                     |
+-----+------------------------+------------------------+
|    10 | dejavu-lgc-sans-fonts        | dejavu-fonts                      |
+-----+------------------------+------------------------+
|    11 | dejavu-sans-fonts               | dejavu-fonts                      |
+-----+------------------------+------------------------+
|    12 | dejavu-serif-fonts             | dejavu-fonts                      |
+-----+------------------------+------------------------+
|    13 | festival-docs                     | festival-help                     |
+-----+------------------------+------------------------+
|    14 | fontawesome-fonts-web        | fontawesome-fonts               |
+-----+------------------------+------------------------+
|    15 | ghostscript-doc                 | ghostscript-help                |
+-----+------------------------+------------------------+
|    16 | glib2-doc                         | glib2-help                        |
+-----+------------------------+------------------------+
|    17 | glibmm24-doc                     | glibmm24-help                    |
+-----+------------------------+------------------------+
|    18 | gnome-getting-started-docs-cs | gnome-getting-started-docs-help |
+-----+------------------------+------------------------+
|    19 | gnome-getting-started-docs-de | gnome-getting-started-docs-help |
+-----+------------------------+------------------------+
|    20 | gnome-getting-started-docs-es | gnome-getting-started-docs-help |
+-----+------------------------+------------------------+
|    21 | gnome-getting-started-docs-fr | gnome-getting-started-docs-help |
+-----+------------------------+------------------------+
|    22 | gnome-getting-started-docs-gl | gnome-getting-started-docs-help |
+-----+------------------------+------------------------+
```

| 23 | gnome-getting-started-docs-hu | gnome-getting-started-docs-help |

+-----+-------------------------+-------------------------+

| 24 | gnome-getting-started-docs-it | gnome-getting-started-docs-help |

+-----+-------------------------+-------------------------+

| 25 | gnome-getting-started-docs-pl | gnome-getting-started-docs-help |

+-----+-------------------------+-------------------------+

| 26 | gnome-getting-started-docs-pt_BR | gnome-getting-started-docs-help |

+-----+-------------------------+-------------------------+

| 27 | gnome-getting-started-docs-ru | gnome-getting-started-docs-help |

+-----+-------------------------+-------------------------+

| 28 | gnome-icon-theme-legacy | gnome-icon-theme |

+-----+-------------------------+-------------------------+

| 29 | gnome-shell-extension-common | gnome-shell-extensions |

+-----+-------------------------+-------------------------+

| 30 | google-noto-emoji-color-fonts | google-noto-emoji-fonts |

+-----+-------------------------+-------------------------+

| 31 | gspell-doc | gspell-devel |

+-----+-------------------------+-------------------------+

| 32 | gssdp-docs | gssdp-help |

+-----+-------------------------+-------------------------+

| 33 | gstreamer-devel-docs | gstreamer-help |

+-----+-------------------------+-------------------------+

| 34 | gstreamer1-devel-docs | gstreamer1-help |

+-----+-------------------------+-------------------------+

| 35 | gstreamer1-plugins-base-devel-docs | gstreamer1-plugins-base-help |

+-----+-------------------------+-------------------------+

| 36 | gtkmm30-doc | gtkmm30 |

+-----+-------------------------+-------------------------+

| 37 | gupnp-docs | gupnp-help |

+-----+-------------------------+-------------------------+

| 38 | hunspell-en-US | hunspell-en |

39	hyphen-en	hyphen-devel
40	ibus-devel-docs	ibus-help
41	keybinder3-doc	keybinder3-devel
42	langtable-python	python2-langtable
43	libX11-common	libX11
44	libdbusmenu-doc	libdbusmenu-help
45	libical-glib-doc	libical-devel
46	liblouis-python	python2-louis
47	libproxy-python	python2-libproxy
48	libsigc++20-doc	libsigc++20-help
49	libssh2-docs	libssh2-help
50	man-pages	man-pages-help
51	ncurses-term	ncurses
52	orc-doc	orc-help
53	p11-kit-doc	p11-kit-devel

| 54 | pangomm-doc | pangomm-help |

+-----+-----+-----+

| 55 | paratype-pt-sans-caption-fonts | paratype-pt-sans-fonts |

+-----+-----+-----+

| 56 | pcp-doc | pcp-help |

+-----+-----+-----+

| 57 | perl-URI | perl-URI-help |

+-----+-----+-----+

| 58 | perl-libwww-perl | perl-libwww-perl-help |

+-----+-----+-----+

| 59 | po4a | po4a-help |

+-----+-----+-----+

| 60 | podman-docker | podman-help |

+-----+-----+-----+

| 61 | polkit-docs | polkit-devel |

+-----+-----+-----+

| 62 | pygtk2-doc | pygtk2-help |

+-----+-----+-----+

| 63 | pyserial | python2-pyserial |

+-----+-----+-----+

| 64 | python-augeas | python2-augeas |

+-----+-----+-----+

| 65 | python-babel | python2-babel |

+-----+-----+-----+

| 66 | python-backports-ssl_match_hostname | python2-backports-ssl_match_hostname |

+-----+-----+-----+

| 67 | python-beaker | python2-beaker |

+-----+-----+-----+

| 68 | python-cherrypy | python2-cherrypy |

+-----+-----+-----+

| 69 | python-configobj | python2-configobj |
+-----+-----+-----+
| 70 | python-configshell | python2-configshell |
+-----+-----+-----+
| 71 | python-dateutil | python2-dateutil |
+-----+-----+-----+
| 72 | python-decorator | python2-decorator |
+-----+-----+-----+
| 73 | python-dns | python2-dns |
+-----+-----+-----+
| 74 | python-docutils | python2-docutils |
+-----+-----+-----+
| 75 | python-enum34 | python2-enum34 |
+-----+-----+-----+
| 76 | python-flask | python2-flask |
+-----+-----+-----+
| 77 | python-hwdata | python2-hwdata |
+-----+-----+-----+
| 78 | python-iniparse | python2-iniparse |
+-----+-----+-----+
| 79 | python-inotify | python2-inotify |
+-----+-----+-----+
| 80 | python-jsonpatch | python2-jsonpatch |
+-----+-----+-----+
| 81 | python-jsonpointer | python2-jsonpointer |
+-----+-----+-----+
| 82 | python-jwt | python2-jwt |
+-----+-----+-----+
| 83 | python-kitchen | python2-kitchen |
+-----+-----+-----+
| 84 | python-linux-procfs | python2-linux-procfs |

85	python-magic	python2-magic
86	python-mako	python2-mako
87	python-memcached	python2-memcached
88	python-netaddr	python2-netaddr
89	python-nose	python2-nose
90	python-ntplib	python2-ntplib
91	python-paramiko	python2-paramiko
92	python-paste	python2-paste
93	python-ply	python2-ply
94	python-py	python2-py
95	python-pycparser	python2-pycparser
96	python-pygments	python2-pygments
97	python-pyudev	python2-pyudev
98	python-requests	python2-requests
99	python-rtslib	python2-rtslib

| 100 | python-setuptools | python2-setuptools |
+-----+-----+-----+
| 101 | python-six | python2-six |
+-----+-----+-----+
| 102 | python-slip | python2-slip |
+-----+-----+-----+
| 103 | python-slip-dbus | python2-slip |
+-----+-----+-----+
| 104 | python-slip-gtk | python2-slip-gtk |
+-----+-----+-----+
| 105 | python-sphinx | python2-sphinx |
+-----+-----+-----+
| 106 | python-sssdconfig | python2-sssd |
+-----+-----+-----+
| 107 | python-suds | python2-suds |
+-----+-----+-----+
| 108 | python-tempita | python2-tempita |
+-----+-----+-----+
| 109 | python-urlgrabber | python2-urlgrabber |
+-----+-----+-----+
| 110 | python-urllib3 | python2-urllib3 |
+-----+-----+-----+
| 111 | python-virtualenv | python2-virtualenv |
+-----+-----+-----+
| 112 | python-webob | python2-webob |
+-----+-----+-----+
| 113 | python-webtest | python2-webtest |
+-----+-----+-----+
| 114 | python-which | python2-which |
+-----+-----+-----+
| 115 | python2-pyasn1-modules | python2-pyasn1 |

116	pytz	python2-pytz
117	qt-doc	qt
118	qt5-qtenginio-doc	qt5-qtenginio-help
119	rhnlib	python2-rhnlib
120	samyak-oriya-fonts	samyak-odia-fonts
121	selinux-policy-doc	selinux-policy-help
122	sendmail-cf	sendmail
123	system-config-printer-libs	system-config-printer
124	tracker-docs	tracker-help
125	wayland-doc	wayland-help
126	xml-common	sgml-common
127	xorg-x11-fonts-misc	xorg-x11-fonts
128	xorg-x11-server-source	xorg-x11-server

4.建议

4.1 可使用特征

name:软件包的名称。

选择理由：软件包的名称通常反映了其核心功能或用途。在很多情况下，软件包的名称就是其功能的一个直观标识。例如，名称中包含 nginx、httpd 等的包很明显指向 Web 服务器功能，因此名称相似的软件包在功能上也可能相似。

provides：软件包提供的功能或服务。

选择理由：是软件包的核心功能标识。它描述了软件包所实现的服务或功能，可能是某个 API、工具集、库等的集合。相似的 provides 字段意味着这两个软件包提供了相同或相似的功能，因此可以作为重要的映射依据。尤其是在跨平台映射中，很多包会标明提供某个标准接口或服务。

requires：软件包的依赖关系

选择理由：软件包的依赖关系反映了它所依赖的外部工具、库或框架。相似的依赖关系通常意味着两个软件包在功能实现上有很多共同点。例如，两个软件包都可能依赖同样的数据库、编程语言运行时或其他关键库。如果它们依赖相同的工具或库，这可能意味着它们解决的问题领域相似。

使用的特征越多，理论上相似度的计算效果越好，同学们可以使用更多的特征

4.2 常见的相似度计算方法

在准备好每个软件包的一系列特征字典后，需要针对每个特征分别调用相似度计算函数，再分别按照一定的权重进行加权求和，以此算出两个软件包的总相似度。本部分将给出一些计算单个特征的相似度时可用的计算方法和代码供大家参考，实验中需要同学们根据不同特征的类型和性质选择合适的相似度计算方法，也鼓励大家尝试其他计算方法来获得更好的结果。

余弦相似度

余弦相似性通过测量两个向量的夹角的余弦值来度量它们之间的相似性。

两个向量间的余弦值可以通过使用欧几里得点积公式求出：

给定两个属性向量 A 和 B，其余弦相似性为：

和　　分别代表向量A和B的各分量。

对于文本匹配，属性向量 A 和 B 通常是文档中的词频向量。余弦相似性可以被看作是在比较过程中把文件长度正规化的方法。余弦相似度适合用于比较文本之间的语义相似度。

基于余弦相似度计算相似度的示例代码如下：

```
import re
from collections import Counter
import math

def cosine_similarity(str1, str2):
    def text_to_vector(text):
        # 使用正则表达式匹配所有单词，并转换为小写，确保大小写不影响比较结果
        words = re.findall(r'\w+', text.lower())
        return Counter(words)

    def dot_product_vectors(vector1, vector2):        # 找到两个向量中共有的键（单词），并计算这些键对应的值（词频）的乘积之和
        common_keys = set(vector1.keys()) & set(vector2.keys())
        return sum(vector1[key] * vector2[key] for key in common_keys)

    def vector_magnitude(vector):
        # 向量各分量平方和的平方根
        return math.sqrt(sum(value ** 2 for value in vector.values()))

    vector1 = text_to_vector(str1)
    vector2 = text_to_vector(str2)

    # 模长
    magnitude1 = vector_magnitude(vector1)
    magnitude2 = vector_magnitude(vector2)

    if magnitude1 == 0 or magnitude2 == 0:
        return 0.0

    # 点积
    dot_product = dot_product_vectors(vector1, vector2)
    # 余弦相似度
    similarity = dot_product / (magnitude1 * magnitude2)

    return similarity
```

Jaccard 相似系数

雅卡尔指数（Jaccard index），又称为交并比（Intersection over Union）、雅卡尔相似系数（Jaccard similarity coefficient），是用于比较样本集的相似性与多样性的统计量。雅卡尔系数能够量度有限样本集合的相似度，其定义为两个集合交集大小与并集大小之间的比例：

如果 A 与 B 完全重合，则定义 J(A,B) = 1。于是有

Jaccard 相似系数是比较集合之间相似度的常见度量方法。

示例代码如下：

```
def jaccard_similarity(str1, str2):
    def get_word_set(text):
        words = set(text.split())
        return words

    set1 = get_word_set(str1)
    set2 = get_word_set(str2)

    intersection = len(set1.intersection(set2))
    union = len(set1.union(set2))

    if union == 0:
        # 如果两个集合都为空，定义相似度为 1
        similarity = 1.0
    else:
        similarity = intersection / union

    return similarity
```

基于 Levenshtein 距离计算相似度

莱文斯坦距离（英语：Levenshtein distance）是编辑距离的一种。指两个字符串之间，由一个转成另一个所需的最少编辑操作次数。

允许的编辑操作包括：

1. 将一个字符替换成另一个字符
2. 插入一个字符
3. 删除一个字符

动态规划经常被用来作为这个问题的解决手段之一。基于 Levenshtein 距离计算相似度的示例代码如下：

```
def levenshtein_similarity(str1, str2):
    len1 = len(str1)
    len2 = len(str2)
    dp = [[0] * (len2 + 1) for _ in range(len1 + 1)]
    for i in range(len1 + 1):
        dp[i][0] = i
    for j in range(len2 + 1):
        dp[0][j] = j
    for i in range(1, len1 + 1):
        for j in range(1, len2 + 1):
            if str1[i - 1] == str2[j - 1]:
                dp[i][j] = dp[i - 1][j - 1]
            else:
                dp[i][j] = min(dp[i - 1][j - 1], dp[i][j - 1], dp[i - 1][j]) + 1
    similarity = 1 - (dp[len1][len2] / max(len1, len2))
    return similarity
```

simhash 算法

SimHash（Similarity Hashing）是一种用于快速估算文本相似度的哈希算法。它通过将高维向量映射为固定长度的二进制串来表示文本，进而可以通过计算这些二进制串之间的汉明距离来衡量文本间的相似性。相似的文档在经过 SimHash 处理后会得到相近的二进制哈希值。

汉明距离（英语：Hamming distance）是两个字符串对应位置的不同字符的个数，也就是将一个字符串变换成另外一个字符串所需要替换的字符个数。

SimHash 常用于大规模文本去重、相似内容检测等场景中，因为其能够高效地估算两个文本之间的相似度，计算汉明距离非常快。

示例代码如下：

```python
import re
import hashlib

def simhash_similarity(str1, str2):
    def get_simhash(text):
        # 预处理文本，移除所有非字母数字字符并转为小写
        processed_text = re.sub(r'[^a-zA-Z0-9]', '', text.lower())
        # 使用 MD5 算法生成文本的哈希值，并将其转换为十六进制字符串
        hash_value = hashlib.md5(processed_text.encode('utf-8')).hexdigest()
        # 将十六进制哈希值转换为二进制字符串，长度为 128 位
        binary_hash = bin(int(hash_value, 16))[2:].zfill(128)

        return binary_hash

    def calculate_hamming_distance(hash1, hash2):
        # 通过比较对应位置上的字符来计算汉明距离
        distance = sum(bit1 != bit2 for bit1, bit2 in zip(hash1, hash2))
        return distance

    # Simhash 值
    simhash1 = get_simhash(str1)
    simhash2 = get_simhash(str2)

    # Simhash 值之间的汉明距离
    distance = calculate_hamming_distance(simhash1, simhash2)

    # 计算相似度，范围在 0 到 1 之间。汉明距离越小，相似度越高
    similarity = 1 - distance / 128

    return similarity
```

使用 difflib 模块

difflib 来自 python 标准库，该模块提供用于比较序列的类和函数。可以使用 difflib.SequenceMatcher 来计算相似度。它主要关注序列之间的共同元素数量和相对顺序，基于最长公共子序列（LCS）。

> 这是一个灵活的类，可用于比较任何类型的序列对，只要序列元素为 hashable 对象。其基本算法要早于由 Ratcliff 和 Obershelp 于 1980 年代末期发表并以“格式塔模式匹配”的夸张名称命名的算法，并且更加有趣一些。 其思路是找到不包含“垃圾”元素的最长连续匹配子序列；所谓“垃圾”元素是指其在某种意义上没有价值，例如空白行或空白符。 （处理垃圾元素是对 Ratcliff 和 Obershelp 算法的一个扩展。） 然后同样的思路将递归地应用于匹配序列的左右序列片段。 这并不能产生最小编辑序列，但确实能产生在人们看来“正确”的匹配。
>
> 耗时：基本 Ratcliff-Obershelp 算法在最坏情况下为立方时间而在一般情况下为平方时间。 SequenceMatcher 在最坏情况下为平方时间而在一般情况下的行为受到序列中有多少相同元素这一因素的微妙影响；在最佳情况下则为线性时间。
>
> 自动垃圾启发式计算：SequenceMatcher 支持使用启发式计算来自动将特定序列项视为垃圾。 这种启发式计算会统计每个单独项在序列中出现的次数。 如果某一项（在第一项之后）的重复次数超过序列长度的 1% 并且序列长度至少有 200 项，该项会被标记为“热门”并被视为序列匹配中的垃圾。 这种启发式计算可以通过在创建 SequenceMatcher 时将 autojunk 参数设为 False 来关闭。

使用 difflib 计算相似度的示例代码如下：

```
import difflib


def difflib_similarity(str1, str2):
    # 使用 difflib.SequenceMatcher 计算两个字符串的相似度比例
    # 第一个参数为 None 表示不使用任何特定的匹配函数，直接比较输入的字符串
    similarity = difflib.SequenceMatcher(None, str1, str2).ratio()

    return similarity
```

4.3 代码框架与 checklist

本实验提供了代码框架供同学们参考。该框架仅启发同学们的思路，不提供可运行

源代码，请同学们自行编写代码。checklist 为挑选后的有效映射关系，同学们最终的映射计算结果应当包含该 list 中的所有内容。