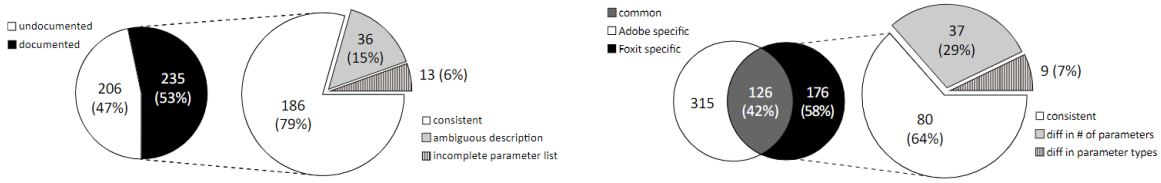
RQ2: Type Inconsistancy

This experiment is in the `Section V-C Inconsistency of Binding Calls` of our paper. The goal of this experiment is to

1. compare the type information of Adobe Reader inferred by TYPEORACLE with the type information inferred by Adobe API Manual;
2. compare the type information of Adobe Reader inferred by TYPEORACLE with the type information of Foxit Reader inferred by TYPEORACLE.

The result is shown in the `Fig. 6: Systematic study of binding call inconsistency` in our paper. We can see from Figure 6a that only 53% of Adobe Reader's binding calls are documented, 21% of which have type inconsistency with the documentation. Further inspection reveals that the inconsistency comes from ambiguous description and incomplete parameter list in the documentation. Figure 6b says that Foxit Reader shares 42% of binding calls with Adobe Reader, 36% of which differ in either the number of parameters or the type of certain parameters.



(a) between documentation and implementation (b) between Adobe Reader and Foxit Reader

Fig. 6: Systematic study of binding call inconsistency.

folder structure

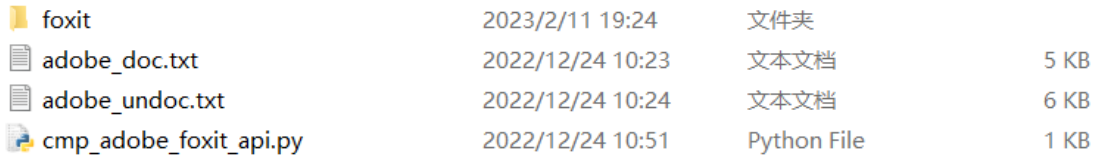
This following are the file names and their descriptions.

```
cmp_adobe_foxit: compare the type information of Adobe with  the type information
of  Foxit

cmp_doc: compare the type information of TypeOracle with  the type information of
Manual

compare_api:
  - cmp_adobe_foxit_api.py: script to get foxit specific APIs and common APIs

adobe_doc.txt: APIs present in Adobe API Manual

adobe_undoc.txt: APIs not present in Adobe API Manual

difference_adobe_foxit.txt: the different APIs in Adobe Reader and Foxit Reader,
including APIs that are different in parameters and APIs that are different in
parameters' types

difference_manual.txt: the different APIs in TypeOracle and Adobe Manual in
Adobe's API list, including APIs that have ambiguous description and incomplete
parameter
```

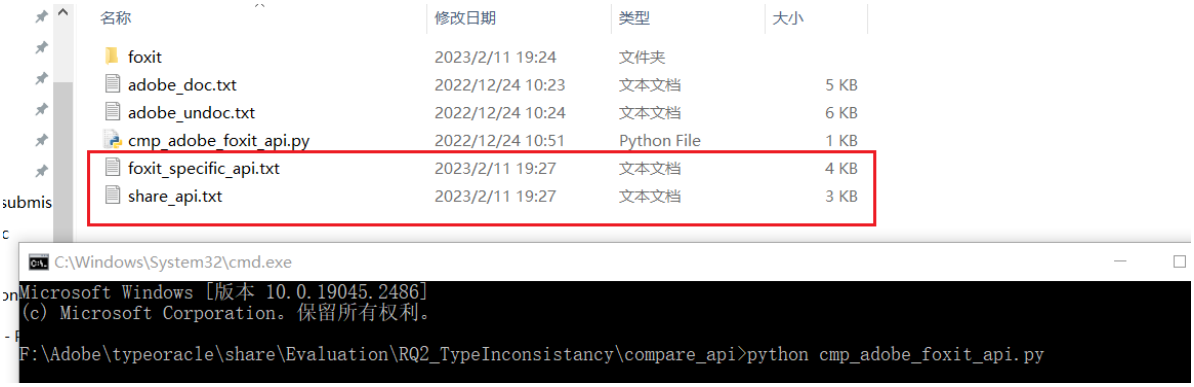
```
foxit_specific_api.txt: APIs that only implemented in Foxit Reader

share_api.txt: APIs that both implemented in Adobe Reader and Foxit Reader
```

How to reproduce

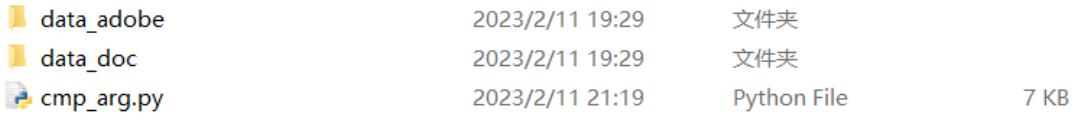
1. prepare these files: adobe_doc.txt, adobe_undoc.txt, funclst.txt(foxit), setterlst.txt(foxit)



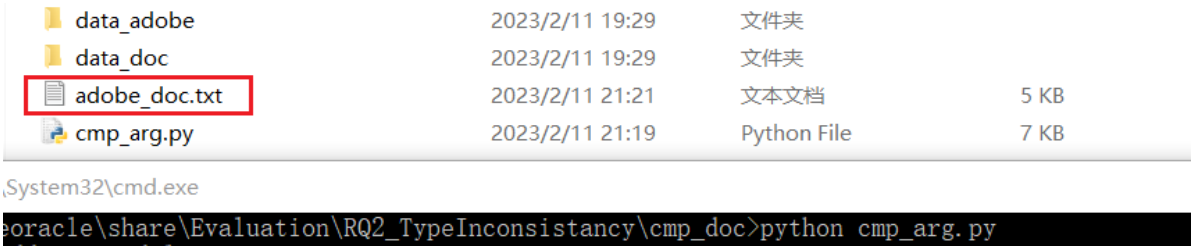
2. execute `cmp_adobe_foxit_api.py` to get foxit specific APIs and common APIs



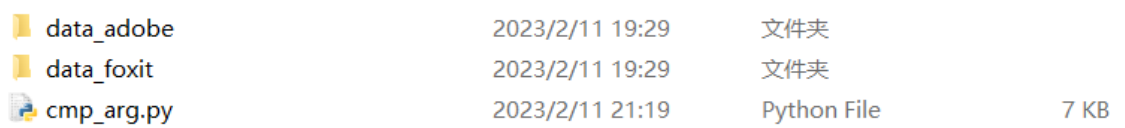
3. compare the type information of TypeOracle with the type information of Manual



4. execute cmp_arg.py to get result in `adobe_doc.txt`



5. compare the type information of Adobe with the type information of Foxit



6. execute cmp_arg.py to get result in `adobe_foxit.txt`