

Automatically generation of shell-driven TFLs with R

Hang Sun, Yi Yang, and Yu Li, Jiangsu Hengrui Pharmaceuticals Co., Ltd.

ABSTRACT

In pharmaceutical industry, programmers usually use data driven method to generate TFLs (tables, listings and figures). But manually reading shells and converting demands from shell to SAS programs are always time-consuming and prone to error.

This paper will provide an efficient way by using r package officer to extract contents from standard shell into SAS dataset. then by converting the dataset into project-level metadata, we can extract key words and phrases and thus generate SAS program macro parameters and statements to output tables and listings.

INTRODUCTION

OFFICER is a powerful R package for creating and updating Microsoft Word (.docx) files. This lets R users manipulate Word (.docx) and PowerPoint (*.pptx) documents. In short, one can add images, tables and text into documents from R. An initial document can be provided; contents, styles and properties of the original document will then be available. It also supports the writing of 'RTF' documents.

In line with all project-level programs, all code is processed within the SAS environment. Since SAS does not have the capability to directly read Microsoft Word (.docx) files, we will use the R package "OFFICER" to assist us in reading mock shells within the SAS environment.

TFL SHELL AND LAYOUT OF THE METADATA

The layout of the TFL shell is crucial for text extraction. The code in this paper is based on the standardized TFL shell. Figure 1 illustrates a portion of the TFL shell. Our task is to extract the keywords into project-level metadata, as shown in Figure 2.

表 14.3.1-1.1 治疗期间出现的不良事件（TEAE）汇总 （安全性分析集）

	治疗组 A (N = xx)
	n (%)
治疗期间出现的不良事件（TEAE）	xx (xx%)
轻度	xx (xx%)
中度	xx (xx%)
重度	xx (xx%)
治疗相关不良事件（TRAE）	xx (xx%)
轻度	xx (xx%)
中度	xx (xx%)
重度	xx (xx%)
导致暂停用药或永久停药的 TEAE	xx (xx%)
导致暂停用药或永久停药的 TRAE	
治疗期间出现的严重不良事件（SAE）	xx (xx%)
治疗相关 SAE	xx (xx%)
导致死亡的 TEAE	xx (xx%)
导致死亡的 TRAE	xx (xx%)
特别关注的不良事件（AESI）	xx (xx%)

注：n=至少发生一次不良事件的受试者数。如果同一受试者发生多次事件，则只计算一次；百分比计算以 N 为分母；只汇总 TEAE 中的 AESI。
TEAE 指所有发生在试验用药品首次用药后至安全随访期结束的不良事件，无论其是否与给药相关；TRAE 指与试验用药品相关性为“肯定有
关”、“可能有关”、“无法判定”的 TEAE。

Figure 1. TFL Shell Layout

TEXT	MASK	LINE_BREAK	INDENT	FILTER			
治疗期间出现的不良事件（TEAE）				(trtemfl="Y")			
轻度				(trtemfl="Y") @ find(aesev, "轻")			
中度				(trtemfl="Y") @ find(aesev, "中")			
重度				(trtemfl="Y") @ find(aesev, "重")			
治疗相关不良事件（TRAE）				(trtemfl="Y") and prxmatch("/肯定有关 可能有关 无法判定/", catx(" ",			
轻度				(trtemfl="Y") and prxmatch("/肯定有关 可能有关 无法判定/", catx(" ",			
中度				(trtemfl="Y") and prxmatch("/肯定有关 可能有关 无法判定/", catx(" ",			
重度				(trtemfl="Y") and prxmatch("/肯定有关 可能有关 无法判定/", catx(" ",			
导致暂停用药或永久停药的TEAE				(trtemfl="Y") and prxmatch("/暂停用药 永久停药/", catx(" ", of aeacr			
导致暂停用药或永久停药的TRAE				(trtemfl="Y") and prxmatch("/暂停用药 永久停药/", catx(" ", of aeacr			
治疗期间出现的严重不良事件（SAE）				(trtemfl="Y") and (aeser="是")			
治疗相关SAE				(trtemfl="Y") and (aeser="是") and prxmatch("/肯定有关 可能有关 无法			
导致死亡的TEAE				(trtemfl="Y") and (aeout="死亡" or aesdth="是")			
导致死亡的TRAE				(trtemfl="Y") and (aeout="死亡" or aesdth="是") and prxmatch("/肯定			
特别关注的不良事件（AESI）				(trtemfl="Y") and (aes="是")			

Figure 2. An example of generated metadata

SAS CODE FOR EXTRACTING TABLE LAYOUT

Following I will introduce the implementation strategy and specific code of this tool step by step.

1.INSTALL AND IMPORT R Package OFFICER

The installation of R packages is straightforward. All R packages used in this tool are given below:

```
/*install packages*/
install.packages("officer")
library("officer")
```

2. DEFINE THE PATH AND FILE NAME OF TFL SHELL

Path of the TFL shell can be stored in a macro in setup documentation (and this file must exist):

```
/*define file path*/
%let tfl_shell_docx ='C:\Users\Administrator\Desktop\test\SHELL.docx';
```

3. READ THE CONTENT OF TFL SHELL

After installing needed packages and setting up the environment, we start to read the content of TFL shell. You should first confirm that the document is clean and stable. There are no annotations and the title numbers are not automatically numbered. R package is used to read in shell, then we use SAS codes to process the data.

```
/* Use R to read Shell*/
proc iml;
    submit tfl_shell_docx / r;
    library(officer)
    shell <- officer::read_docx("YOUR_TFL_SHELL_NAME")
    r_tfl_shell <- docx_summary(shell)
    call ImportDataSetFromR("TFL_SHELL_SAS_DATASET", "r_tfl_shell");
    endsubmit;
quit;
```

After executing the code above, we can import text file and convert into a list, as shown in the table below:

⊕	ROW	△	TEXT
1	1		治疗期间出现的不良事件 (TEAE)
2	2		轻度
3	3		中度
4	4		重度
5	5		治疗相关不良事件 (TRAE)
6	6		轻度
7	7		中度
8	8		重度
9	9		导致暂停用药或永久停药的TEAE
10	10		导致暂停用药或永久停药的TRAE
11	11		治疗期间出现的严重不良事件 (SAE)
12	12		治疗相关SAE
13	13		导致死亡的TEAE
14	14		导致死亡的TRAE
15	15		特别关注的不良事件 (AESI)

Table 1. texts read from shell

4. COLLATE THE CONTENTS OF THE LIST, CONVERT IT INTO DATA FRAME, AND OUTPUT IT INTO METADATA

In step 3, the texts are read in and stored in a list (see Table 1). Next, we build the conditions and match filters to complete the entire metadata. After processing, all texts and filter conditions are stored in the metadata (see Table 2). As shown in Figure 3, the texts correspond to each row in the shell, and filters are automatically generated with designated frames to match the required condition of each row. We can then output this metadata as illustrated in Figure 2.

Building up conditions can be flexible across various TFLs and projects. We can adjust the matching code based on the specific requirements of different projects and TFLs.

The codes are as follows:

```

/* build up the conditions */

%ae_filter(dsnin= TFL_SHELL_SAS_DATASET, dsnout=
MODIFIED_TFL_SHELL_SAS_DATASET)

/* export the shell and conditions to an external xlsx file */

proc export data= MODIFIED_TFL_SHELL_SAS_DATASET
outfile="&tfl_shell_xlsx" dbms=xlsx;

run;
```

⊕ ROW	🚩 TEXT	🚩 FILTER
1	1 治疗期间出现的不良事件 (TEAE)	(trtemfl="Y")
2	2 轻度	(trtemfl="Y") @ find(aesev, "轻")
3	3 中度	(trtemfl="Y") @ find(aesev, "中")
4	4 重度	(trtemfl="Y") @ find(aesev, "重")
5	5 治疗相关不良事件 (TRAE)	(trtemfl="Y") and prxmatch("/肯定有关 可能有关 无法判定/", catx(" ", of aer...
6	6 轻度	(trtemfl="Y") and prxmatch("/肯定有关 可能有关 无法判定/", catx(" ", of aer...
7	7 中度	(trtemfl="Y") and prxmatch("/肯定有关 可能有关 无法判定/", catx(" ", of aer...
8	8 重度	(trtemfl="Y") and prxmatch("/肯定有关 可能有关 无法判定/", catx(" ", of aer...
9	9 导致暂停用药或永久停药的TEAE	(trtemfl="Y") and prxmatch("/暂停用药 永久停药/", catx(" ", of aeacn:))
10	10 导致暂停用药或永久停药的TRAE	(trtemfl="Y") and prxmatch("/暂停用药 永久停药/", catx(" ", of aeacn:)) and...
11	11 治疗期间出现的严重不良事件 (SAE)	(trtemfl="Y") and (aeser="是")
12	12 治疗相关SAE	(trtemfl="Y") and (aeser="是") and prxmatch("/肯定有关 可能有关 无法判...
13	13 导致死亡的TEAE	(trtemfl="Y") and (aeout="死亡" or aesdth="是")
14	14 导致死亡的TRAE	(trtemfl="Y") and (aeout="死亡" or aesdth="是") and prxmatch("/肯定有...
15	15 特别关注的不良事件 (AESI)	(trtemfl="Y") and (aesl="是")

Table 2. Metadata - texts and filter conditions in corresponding relationship

5. GENERATE OUTPUT TFL

After the metadata is generated, we can automatically create programs based on the matched phrases to produce the required TFLs. Figure 3 provides an example of the above metadata generated TFL.

ABC-1234-101 CSR TFL		Page 1 of 1
表 14.3.1-1.1 治疗期间出现的不良事件 (TEAE) 汇总 (安全性分析集)		
	ABC-1234 (N = 26)	
	n (%)	
治疗期间出现的不良事件 (TEAE)	6 (23.1)	
轻度	5 (19.2)	
中度	1 (3.8)	
重度	0 (0.0)	
治疗相关不良事件 (TRAE)	2 (7.7)	
轻度	1 (3.8)	
中度	1 (3.8)	
重度	0 (0.0)	
导致暂停用药或永久停药的TEAE	0 (0.0)	
导致暂停用药或永久停药的TRAE	0 (0.0)	
治疗期间出现的严重不良事件 (SAE)	0 (0.0)	
治疗相关SAE	0 (0.0)	
导致死亡的TEAE	0 (0.0)	
导致死亡的TRAE	0 (0.0)	
特别关注的不良事件 (AESI)	0 (0.0)	

Figure 3. Generated TFL – AE Summary Table

ISSUES WE HAVE ENCOUNTERED

Here we will share some issues we have encountered in our daily work and corresponding solutions.

To save time and space, the code used above does not read the shell each time you run your programs. Once the metadata is generated, subsequent program runs will skip the reading steps. Therefore, if we update the shell, we would either manually adjust the content or delete the old metadata and regenerate the entire metadata again.

CONCLUSION

In conclusion, this paper has presented an implementation strategy using the TFL shell for efficient text extraction and metadata generation. By leveraging automated processes and flexible matching mechanisms, we have demonstrated how to streamline the creation of project-level metadata from textual data. The methodology outlined not only enhances efficiency by minimizing redundant operations but also ensures adaptability across different projects and datasets. Future work could explore further automation enhancements and integration with advanced analytics for deeper insights. Overall, this approach underscores the importance of structured metadata in facilitating data-driven decision-making and operational efficiency in project management.

REFERENCES

R Basics. (2024, February 7). *The Ultimate Guide to the officer Package in R*.

<https://rbasics.org/packages/officer-package-in-r/#:~:text=What%20is%20the%20officer%20package%3F%20In%20this%20section%2C,automated%20and%20reproducible%20reports%20within%20the%20R%20environment.>

officer package - RDocumentation. (n.d.).

<https://www.rdocumentation.org/packages/officer/versions/0.6.5>

ACKNOWLEDGMENTS

I'd like to thank all my colleagues for their useful feedbacks and suggestions when using this tool. Thank Yi Yang and Yu Li for their support of the ideas and codes.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Hang Sun

Company: Jiangsu Hengrui Pharmaceuticals Co., Ltd.

Phone: 18912804889

Email: hang.sun.hs57@hengrui.com

APPENDIX

SAS Code

```
/*install packages*/
install.packages("officer")
library("officer")

/*define file path*/
%let tfl_shell_docx ='C:\Users\Administrator\Desktop\test\SHELL.docx';

/* Use R to read Shell*/
proc iml;
    submit tfl_shell_docx / r;
    library(officer)
    shell <- officer::read_docx("YOUR_TFL_SHELL_NAME")
    r_tfl_shell <- docx_summary(shell)
    call ImportDataSetFromR("TFL_SHELL_SAS_DATASET", "r_tfl_shell");
    endsubmit;
quit;
/* build up the conditions */
%ae_filter(dsnin= TFL_SHELL_SAS_DATASET, dsnout=
MODIFIED_TFL_SHELL_SAS_DATASET)

/* export the shell and conditions to an external xlsx file */
proc export data= MODIFIED_TFL_SHELL_SAS_DATASET
outfile="%tfl_shell_xlsx" dbms=xlsx;
run;
```