1. 对一些常见的已知的噪声或有特殊意义的编码，直接进行替换

举例：

“ ”直接删除

```
"把一个圆柱形橡皮泥加工成一个和它等底等高的圆锥，体积比原来减少了(    )。",
```

“{#blank#}”和“{#/blank#}”分别为左右括号，大多出现在填空处，替换为“【”和“】”

```
"5后面连续三个数是{#blank#}1{#\/blank#}、
```

“>”和“<”分别代表大于号和小于号

```
"在横线上填上“&gt;”、“&lt;”或“=”。
```

“<”替换为“＜”（在有双引号的情况下，英文符替换为中文符，防止与括号产生歧义）

- 这个处理其实是最后突发奇想添加上去的，因为通过对数据的浏览发现这种噪声是十分常见的甚至会大面积出现，实验结果证明处理确实变得更加精细了，直觉上也确实是噪声越少，处理效果会越好。


```
"stem" : "对的在横线上打“√”，错的在横线上打“×”。\n<p>2+3=5{#blank#}1{#\/blank#}            4-1&gt;3{#blank#}2{#\/blank#}  
         1+1=0{#blank#}3{#\/blank#}<\/p>\n\n<p>6&gt;3+3{#blank#}4{#\/blank#}           
 5&lt;5+0{#blank#}5{#\/blank#}          7+2=9{#blank#}6{#\/blank#}<\/p>\n",
```

```
"stem" : "估算。<br>412×297≈      4.92×0.61≈     27.495÷3.14≈<br>28×39≈          587×8≈   
       6285÷7≈",
```


- 以第二张图的内容为例，在删除这些“ ”前，混杂在大量杂讯中的纯算式几乎无法被识别，删除之后这些算式全部成功保留。

2. 对输入串进行分割

- **分割标准**:
- （1）一串中文（包括符号）的结束/开始
- （2）出现英文符号的括号，将整个括号中的内容作为一个片段(实验表明出现这种情况基本为噪声)
- （3）数字，字母，特殊符号

```
"在横线上填入“&gt;”“&lt;”或“=”。\n<p align=\"left\">57×9{#blank#}1{#\/blank#}59×7     
                              
   4.5元{#blank#}2{#\/blank#}4元5分<\/p>\n\n<p align=\"left\">788-347-234{#blank#}3{#\/blank#}788-（347＋234）
      42÷7-3{#blank#}4{#\/blank#}42÷（7-3）<\/p>\n"
```

分割后→

```
['在横线上填入“>”“<”或“=”。',
'\\n',
'<palign=\"left\">',
'57×9【1】59×74.5元【2】4元5分',
'</p>',
'\\n\\n',
'<palign=\"left\">',
'788-347-234【3】788-（347＋234）42÷7-3【4】42÷（7-3）',
'</p>',
'\\n']
```

3. 提取有用的片段

- 这个部分的做法是比较直觉性的，因为在字符串层面其实做了很多的微调都没有找到比较完美的解决方案。大致思想是：
- （1）对于中文片段直接提取
- （2）对于单独的字母，数字，符号片段，如果出现运算符或与中文片段相连接，也视为题目条件的一部分
- （3）对于英文符括号中的内容，若出现四则运算符或内容为中文则视为题目条件，否则视为杂讯（此处其实并不会出现多少误差，因为通过对数据的浏览发现“<>”中几乎无一例外为杂讯，“[]”几乎不出现，“()”中几乎均为题目条件或填空位置）
- （4）部分特殊符号片段，直接删除

```
['在横线上填入“>”“<”或“=”。',
'\\n',
'<palign=\"left\">',
'57×9【1】59×74.5元【2】4元5分',
'</p>',
'\\n\\n',
'<palign=\"left\">',
'788-347-234【3】788-（347+234）42÷7-3【4】42÷（7-3）',
'</p>',
'\\n']
```

提取后

```
'在横线上填入“>”“<”或“=”。57×9（1）59×74.5元（2）4元5分788-347-234（3）788-（347+234）42÷7-3（4）42÷（7-3）'
```

4. 对运算符，数字字母汉化

- 因为内容已经提取完毕，对运算符或特殊标记进行较简单的词库替换即可，词库目前有51条，后续可以持续添加

- （似乎符号有半角和全角的区别，实验中发现必须把半角的“+”和全角的“+”都加入词库才能全部替换，但在json文件中这两个“+”却被识别为重复的条目，这一点暂不清楚原因）

- 数字和字母直接提取后，数字直接替换为“[数据]”，单个字母替换为“[变量]”，多个字母替换为“[条件量]”（这一点可能也设计语义问题）

```
'在横线上填入“>”“<”或“=”。57×9（1）59×74.5元（2）4元5分788-347-234（3）788-（347+234）42÷7-3（4）42÷（7-3）'
```

替换后→

```
"在横线上填入“大于”“小于”或“等于”。
[数据]乘[数据][空格][数据]乘[数据]元[空格][数据]元[数据]分
[数据]减[数据]减[数据][空格][数据]负（[数据]加[数据]）
[数据]除以[数据]减[数据][空格][数据]除以（[数据]减[数据]）"
```

尚待解决的问题

- （1）符号的二义性，例如“<”可能作为小于号符号也可能作为左括号，g和m等可能作为变量也可能作为单位（目前的方案中若在数字之后则视为单位，否则视为变量），“-”可能为减号也可能为负号，这一点或许要在语义层面进行分析。

- （2）无效题目的删除，一些题目去噪后没有和知识点相关的有效信息了，可能是由于原本的题目条件大多为图片形式。


```
"id": 141030,
"questionid": 43190555,
"content": "填一填。"
```

```
"id": 141027,
"questionid": 43190585,
"content": "解决问题。"
```

- （3）混杂在大量杂讯中的纯字母数字信息可能无法提取。（由于目前杂讯基本类似html语言，有固定格式，较方便分割，如果是其他形式可能需要考虑其他方法）

- （4）汉化过程较简单粗暴，未考虑一些可能出现的特殊情况（例如字母+数字编号作为变量或题目条件或特殊算式）