#2017201998 陈玮婧

“起个好名字”——论文摘要生成标题的研究

一、背景介绍

一篇论文的题目，就像是这篇论文的名片，是读者看到这篇论文时的第一印象。给论文起一个好名字，是十分重要的。对读者来说，他希望看到的题目之后，可以看到他预期中的相应内容；对作者来说，一个好的题目可以吸引读者的兴趣，让他们更愿意仔细去阅读这篇文章。综合来说，就是论文的题目要能清楚展现出这篇论文的亮点和重点。

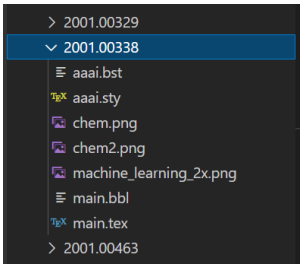
一般来说，论文的题目会在整篇文章基本完成之后，才开始拟定。也就是说我们一般是根据论文的内容来给它起题目。这个过程，可否让机器来完成呢？简化一点，就是提供给计算机一段论文的摘要（摘要一般是整篇论文内容的高度概括），希望它生成贴合文章内容又富有新意的标题。

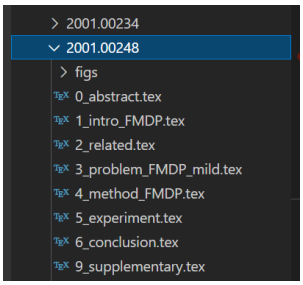
arXiv 是一个收集物理学、数学、计算机科学与生物学论文预印本的网站。很多研究者在论文正式发表之前，会将论文先挂到这个网站上，供网站的其他使用者研读。Arxiv6K 数据集里收录了 6000 多份 arXiv 上论文的 latex 源文件，其中包含文字图片等等，文章在网站上发表的时间跨度从 2001 年到 2012 年，是一个比较丰富的论文数据集。因此，我从 Arxiv6K 数据集中提取出了论文的摘要和题目，构建了一个论文题目摘要数据集，并提出了两种生成模型，在该数据集上进行了一些生成实验，对结果进行了细致分析。

二、数据集构建

1、观察 Arxiv6K 数据集中的数据结构

1）文件结构





可以看到，在每个论文 id 命名的文件夹下，可能存在多个 tex 文件，共同组成了整篇论文的内容。因此这里我选择，遍历每个文件夹下的所有 tex 文件，然后从中去提取论文的题目和摘要。

2）文本结构

标题主要以以下形式存在 latex 文本中：

```
\title{\hspace*{-1ex}The \hspace*{0.1ex}Traveling Observer Model: Multi-task \hspace*{-1ex}\mbox
{Learning \hspace*{-0.15ex}Through \hspace*{-0.1ex}Spatial \hspace*{-0.4ex}Variable \hspace*{-0.05ex}
Embeddings}}
```

即标题的格式主要为：\(……)title{<title>}

摘要主要以以下形式存在 latex 文本中：

```
\section*{Abstract}
The field of Artificial Intelligence (AI)  is going through a period of great expectations, introducing
a certain level of anxiety in research, business and also policy. This anxiety is further energised by
an AI race narrative that makes people believe they might be missing out. Whether real or not,  a
belief in this narrative may be detrimental as some stake-holders will feel obliged to cut corners on
```

```
\begin{abstract}
 Risk management resulting from the actions and states of the different elements making up a operating
 room is a major concern during a surgical procedure. Agent-based simulation shows an interest through
 its interaction concepts, interactivity and autonomy of different simulator entities. We want in our
 study to implement a generator of alerts to listen the evolution of different settings applied to the
 simulator of agents(human fatigue, material efficiency, infection rate ...). This article presents our
 model, its implementation and the first results obtained. It should be noted that this study also made
 it possible to identify several scientific obstacles, such as the integration of different levels of
 abstraction, the coupling of species, the coexistence of several scales in the same environment and
 the deduction of unpredictable alerts. Case-based reasoning (CBR) is a beginning of response relative
 to the last lock mentioned and will be discussed in this paper.
\end{abstract}
```

```
\begin{abstract}

\input{abstract}

\end{abstract}
```

即摘要的格式主要为：① \(……)abstract{
<abstract>
}
② \section{abstract}
<abstract>
③ \begin{abstract}
<abstract>
\end{abstract}
④ input{xxx.tex}，摘要内容在 xxx.tex 文件中

2、提取标题和摘要

在观察完数据结构后，构建论文题目摘要数据集的过程可分为以下 3 个步骤：

1）正确识别出标题和摘要内容部分

之前已经通过观察，得出了题目和摘要在论文的 tex 文件中具体的格式，因此可以直接根据这些格式，使用正则表达式提取出需要的题目和摘要内容。具体的处理方法可以参考代码 get_data.py。

2）去掉设置字体大小等格式的无关词语

由于一些会议还有期刊对论文有严格的排版要求，所以基本上很多论文的题目和摘要中都包含了设置格式的指令等与内容无关的东西，需要将它们剔除出去。它们也都有着比较固定的格式，也可以使用正则表达式将它们筛选出来，进行处理。这里我总结了一些常见的字体设置指令、页面设置指令和其他设置指令：

◆ **字体设置指令**

- 字体族：\textrm{<*text*>} \textsf{<*text*>} \texttt{<*text*>} \textnormal{<*text*>} \text{<*text*>} \rmfamily \sffamily \ttfamily \rm \tt \sf \normalfont
- 字体形状：\textup{<*text*>} \textit{<*text*>} \textsl{<*text*>} \textsc{<*text*>} \upshape \itshape \slshape \scshape \it \sl \sc
- 字体系列：\textmd{<*text*>} \textbf{<*text*>} \bfseries \mdseries \bf \b
- 字体大小：\large \LARGE \Large \huge \Huge \HUGE \small \tiny \normalsize \footnotesize \fontsize{<*size*>}{<*size*>}\selectfont

- 字体效果：\emph{<*text*>} \em \hl \underline \hat \tilde \widetilde \bar \dots
- 数学字体：\mathrm{<*text*>} \mathit{<*text*>} \mathbf{<*text*>} \mathsf{<*text*>} \mathtt{<*text*>} \mathcal{<*text*>} \mathbb{<*text*>} \mathfrak{<*text*>} \mathscr{<*text*>} \cal \ensuremath

◆ **页面设置指令**

- 换行分段：\newline \par
- 对齐设置：\raggedright \raggedleft \centering \begin{center}…\end{center} \begin{flushleft}…\end{flushleft} \begin{flushright}…\end{flushright} \noindent \indent
- 空格间距：\vspace{<size>} \vspace*{<size>} \hspace{<size>} \hspace*{<size>} \hskip{<size>} \vskip{<size>} \medskip \bigskip \smallskip \unskip \quad \qquad \xspace
- 整合排版：\fbox{<*text*>} \mbox{<*text*>} \hbox{<*text*>} \vbox{<*text*>}

◆ **其他设置指令**

- 索引：\gls{<text>} \glspl{<text>} \Gls{<text>} \Glspl{<text>}
- 标签引用：\ref{<text>} \citep{<text>} citet{<text>} \cite{<text>} \footnote{<text>} \blfootnote{<text>} \tnoteref{<text>} \tnotetext{<text>} \label{<text>} \begin{quote}…\end{quote}
- 插入网页：\href{<website>} \url{<website>}
- 致谢：\thank{<text>}

3）将自定义的命令还原成对应的内容

很多论文在写作过程中，会自定义一些命令，实现一些自定义的格式或者进行字符串的替换（例如：指定\CNN 在文中显示为 convolutional neural networks）。因此在构建数据集时，还需要把这部分自定义指令，还原成作者想要表达的真实文本。

自定义指令也有一些固定的格式，总结如下：

- \def{<text>}
- \newcommand<command>[arg num][default value]{<text>}
- \renewcommand<command>[arg num][default value]{<text>}

同样的，可以先用正则表达式识别出它们，然后制作出一个字典，存放指令和它对应的操作，然后在最后，对题目和摘要进行检索，将其中的自定义指令进行替换，从而还原出真正的内容。

3、论文标题摘要数据集格式

将提取并处理过后的标题和摘要按照论文 id，存成一个 json 文件，其中数据的格式为：

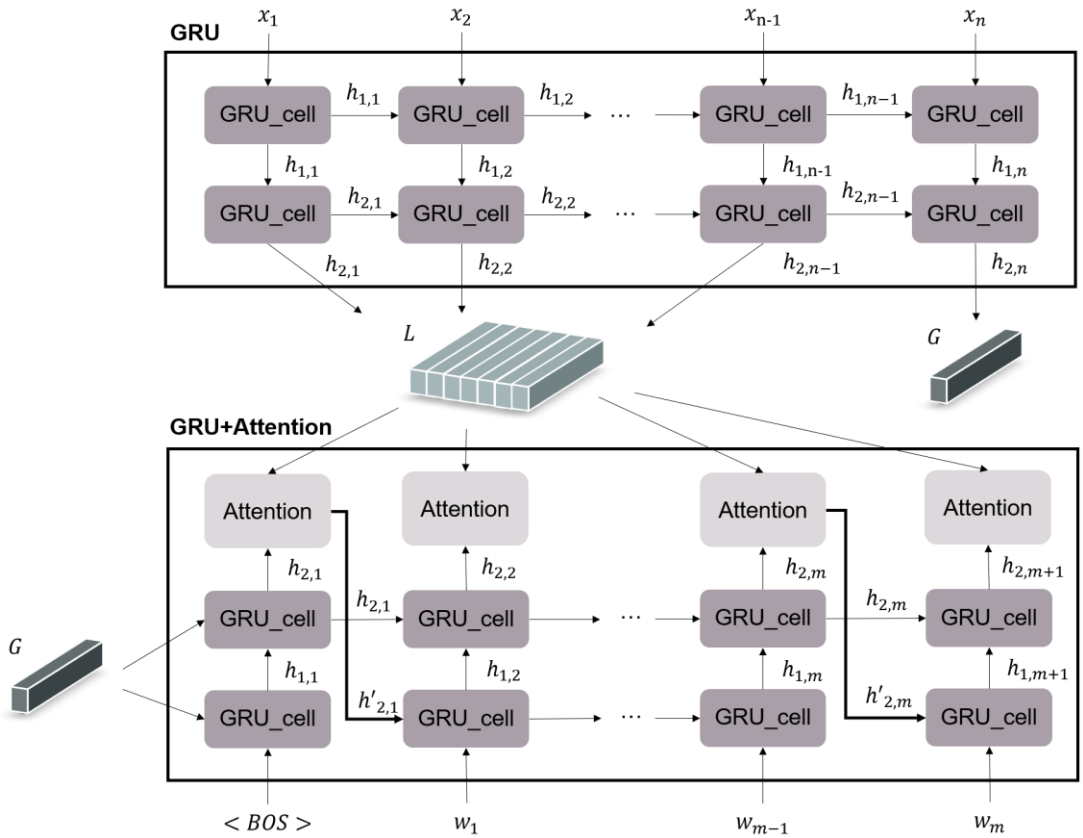
id: {“title”: “……”, “abstract”: “……”}

只需要知道一篇论文在 arxiv 上的 id，根据其检索即可获知其对应的题目和摘要。数据集的具体内容如下图所示：

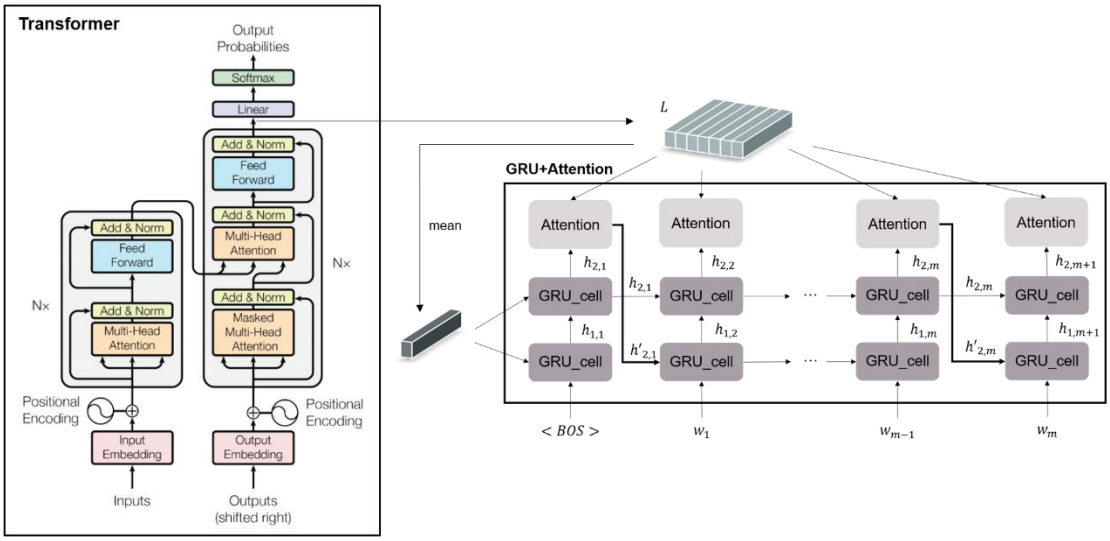
```
"2011.00160": {"title": "Automatic Chronic Degenerative Diseases Identification Using Enteric Nervous
System Images", "abstract": "Studies recently accomplished on the Enteric Nervous System have shown that
chronic degenerative diseases affect the Enteric Glial Cells (EGC) and, thus, the development of
recognition methods able to identify whether or not the EGC are affected by these type of diseases may
be helpful in its diagnoses. In this work, we propose the use of pattern recognition and machine
learning techniques to evaluate if a given animal EGC image was obtained from a healthy individual or
one affect by a chronic degenerative disease. In the proposed approach, we have performed the
classification task with handcrafted features and deep learning based techniques, also known as
non-handcrafted features. The handcrafted features were obtained from the textural content of the ECG
images using texture descriptors, such as the Local Binary Pattern (LBP). Moreover, the representation
learning techniques employed in the approach are based on different Convolutional Neural Network (CNN)
architectures, such as AlexNet and VGG16, with and without transfer learning. The complementarity
between the handcrafted and non-handcrafted features was also evaluated with late fusion techniques. The
datasets of EGC images used in the experiments, which are also contributions of this paper, are composed
of three different chronic degenerative diseases: Cancer, Diabetes Mellitus, and Rheumatoid Arthritis.
The experimental results, supported by statistical analysis, shown that the proposed approach can
distinguish healthy cells from the sick ones with a recognition rate of 89.30 % (Rheumatoid Arthritis),
98.45 % (Cancer), and 95.13 % (Diabetes Mellitus), being achieved by combining classifiers obtained both
feature scenarios."}, "2011.04121": {"title": "Distance-Based Anomaly Detection for Industrial Surfaces
Using Triplet Networks", "abstract": "Surface anomaly detection plays an important quality control role
in many manufacturing industries to reduce scrap production. Machine-based visual inspections have been
```


三、模型设计

针对这一任务，我设计了两种生成标题的模型，一个是基于 GRU 的模型，一个是基于 Transformer+GRU 的模型。它们均采用编码器-解码器架构，输入一段摘要的文本，然后输出一段文本作为这段摘要的题目。具体结构如下图所示：



基于 GRU 的模型结构示意图



基于 Transformer+GRU 的模型结构示意图

四、实验过程与结果分析

1、准备工作

对已经处理好的论文标题摘要数据集，加载到模型上之前还需要对其进行分词、构造字典和切分数据集等工作

1）提取字典

- 将所有文本统一转换为小写
- 去掉标点符号（这里保留了“：”，因为它出现在标题中是有含义的）
- 将 a-b、a/b 类型的词语拆分为词语 a 和词语 b
- 对名词、动词进行词性还原
- 去掉低频词语，控制字典大小（最终在 4500 左右）

2）切割数据集

由于整个数据集大小为 5700 左右，规模较小，为了训练得到更好的模型，这里按照 9:0.5:0.5 的比例切分训练集、验证集和测试集，它们的大小分别为：5172、287、288

2、实验

1、基于 GRU 模型的实验

在构建好的论文标题摘要数据集上进行实验，首先在基于 GRU 的模型上，根据数据集具体情况对参数进行调整。使用 BLEU、METEOR、ROUGE 评测结果，具体如下表所示：

model	Layers	Batch	lr	dropout	BLEU	METEOR	ROUGE
GRU+GRU	1	8	0.0005	0.3	0.152	**0.136**	0.277
	2				**0.177**	0.126	**0.280**
	3				0.130	0.122	0.212
	4				0.103	0.008	0.185

model	Layers	Batch	lr	dropout	BLEU	METEOR	ROUGE
GRU+GRU	2	8	0.0001	0.3	0.143	0.099	0.232
			0.0005		**0.177**	**0.126**	**0.280**
			0.001		0.093	0.070	0.157
				0.2	0.163	0.112	0.257
				0.1	0.163	0.112	0.257
				0.5	0.155	0.111	0.257

可以看到在参数设置合理的情况下，使用 2 层的 GRU 作为编码器和解码器，效果最好。

2、基于 Transformer+GRU 模型的实验

model	Layers	Heads	lr	dropout	BLEU	METEOR	ROUGE
Transformer+ GRU	1+2	8	1e -5	0.3	0.128	0.092	0.218
		4			0.119	0.089	0.209
		2			0.123	0.094	0.204
		1			0.116	0.093	0.202
Transformer+ GRU	2+2	8			0.111	0.080	0.184
	3+2	8			0.146	0.100	0.235
	4+2	8			**0.166**	**0.127**	**0.325**
Transformer (-PE)+GRU	4+2	8			0.133	0.089	0.205

可以看到在参数设置合理的情况下，使用 4 层 Transformer 作为编码器和 2 层的 GRU 作为解码器，效果最好。而且可以看到，去掉 position encoding 之后，结果下降很多，说明输入的顺序是会影响最后标题生成的。综合结果来看，第二种模型比第一种模型要更好。

3、进一步实验

虽然从指标上看，模型生成的结果还可以，但是查看它生成的标题，发现实际的生成效果不佳。生成的标题很单一，而且存在不通顺的情况。为了生成更好的更符合要求的标题，尝试改进：去掉摘要中对生成标题没有帮助的停用词，然后再进行实验，结果如下：

model	Layers	Batch	lr	dropout	BLEU	METEOR	ROUGE
GRU+GRU	1	8	0.0005	0.3	0.115	0.092	0.195
	2				0.089	0.081	0.177
	3				**0.127**	**0.089**	**0.212**
	4				0.048	0.087	0.109

model	Layers	Heads	lr	dropout	BLEU	METEOR	ROUGE
Transformer +GRU	1+2	8	0.0005	0.3	0.150	0.110	0.259
	2+2				0.132	0.104	0.233
	3+2				0.132	0.098	0.243
	4+2				**0.193**	**0.149**	**0.328**

对比前面的实验来看，基于 GRU 的模型没有提升，而基于 Transformer+GRU 的模型提升了一些，在 ROUGE 指标上提升不明显。再肉眼观察生成的题目对比其真正的题目，此改进的实际效果不大。如果想生成更好的题目，还需要进一步改进模型。

五、总结

1、通过一系列方法，提取构造了一个论文题目摘要数据集

2、使用基于 GRU 的模型和基于 Transformer+GRU 的模型，在论文题目摘要数据集上进行了摘要生成题目的任务。

3、不足之处：生成的标题比较单一，而且有的存在不通顺的问题。未来如果想继续深挖更好地实现这一任务目标，可以考虑使用关键词来辅助生成题目，或者在生成题目之前预先给予一些模板供其参考。