



(12)发明专利

(10)授权公告号 CN 104424337 B

(45)授权公告日 2018.03.06

(21)申请号 201310412851.2

(51)Int.Cl.

(22)申请日 2013.09.11

G06F 17/30(2006.01)

(65)同一申请的已公布的文献号

G06F 17/22(2006.01)

申请公布号 CN 104424337 A

(56)对比文件

(43)申请公布日 2015.03.18

CN 101763407 A, 2010.06.30,

(73)专利权人 北大方正集团有限公司

审查员 杜欣威

地址 100871 北京市海淀区成府路298号方

正大厦9层

专利权人 北京方正阿帕比技术有限公司

方正信息产业控股有限公司

(72)发明人 陈聪 郭巍

(74)专利代理机构 北京友联知识产权代理事务  
所(普通合伙) 11343

代理人 尚志峰 汪海屏

权利要求书2页 说明书7页 附图3页

(54)发明名称

文档分割系统和文档分割方法

(57)摘要

B 本发明提供了一种文档分割系统，包括：指令处理单元，用于根据接收到的跳转指令和预设的页面跳转规则，确定当前文档中的第一位置；数据截取单元，用于按照预设的截取范围对所述第一位置附近的文档数据片段进行截取；数据匹配单元，用于将所述文档数据片段与预定义的断点匹配字符进行匹配；文档分割单元，用于在所述文档数据片段中存在与所述断点匹配字符相匹配的数据的情况下，根据该相匹配的数据所处的第二位置分割所述文档，以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。本发明还提出了一种文档分割方法。通过本发明的技术方案，可以针对用户的跳转需求，快速准确地实现文档分割，避免分割处的字符不完整。



1. 一种文档分割系统,其特征在于,包括:

指令处理单元,用于根据接收到的跳转指令和预设的页面跳转规则,确定当前文档中的第一位置;

数据截取单元,用于按照预设的截取范围对所述第一位置附近的文档数据片段进行截取;

数据匹配单元,用于将所述文档数据片段与预定义的断点匹配字符进行匹配;

文档分割单元,用于在所述文档数据片段中存在与所述断点匹配字符相匹配的数据的情况下,根据该相匹配的数据所处的第二位置分割所述文档,以使所述相匹配的数据作为分割得到的后一个文档片段的起始端;

当存在多个所述断点匹配字符时,多个断点匹配字符之间存在优先级,其中,所述数据匹配单元按照优先级从高至低的顺序将多个断点匹配字符依次用于与所述文档数据片段进行匹配,直至获取相匹配的数据;以及

所述文档分割系统还包括:

优先级调整单元,用于在与所述文档数据片段匹配成功的情况下,调整相应的断点匹配字符对应的优先级。

2. 根据权利要求1所述的文档分割系统,其特征在于,所述数据截取单元还用于:在所述文档数据片段中不存在与所述预定义的断点匹配字符相匹配的数据的情况下,则扩大所述截取范围,以重新获取所述文档数据片段;

其中,所述数据匹配单元将重新获取的文档数据片段与所述断点匹配字符进行匹配,若仍不存在,则继续由所述数据截取单元扩大所述截取范围和重新截取所述文档数据片段,直至查找到与所述断点匹配字符相匹配的数据,并由所述文档分割单元根据该相匹配的数据所处的第二位置分割所述文档,以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。

3. 根据权利要求1所述的文档分割系统,其特征在于,所述指令处理单元在所述文档对应的字节数据中确定所述第一位置;

所述数据截取单元在所述文档对应的字节数据中截取所述数据片段;以及

所述数据匹配单元获取所述断点匹配字符对应的字节数据,并在所述文档数据片段中进行匹配,以确定所述第二位置。

4. 根据权利要求3所述的文档分割系统,其特征在于,还包括:

字符转换单元,用于获取所述文档使用的字符集,以用于对所述断点匹配字符进行转换;

其中,所述数据匹配单元利用转换后的断点匹配字符对所述文档数据片段进行匹配。

5. 根据权利要求1至4中任一项所述的文档分割系统,其特征在于,所述断点匹配字符包括标点符号。

6. 一种文档分割方法,其特征在于,包括:

根据接收到的跳转指令和预设的页面跳转规则,确定当前文档中的第一位置;

按照预设的截取范围对所述第一位置附近的文档数据片段进行截取;

若所述文档数据片段中存在与预定义的断点匹配字符相匹配的数据,则根据该相匹配的数据所处的第二位置分割所述文档,以使所述相匹配的数据作为分割得到的后一个文档

片段的起始端；

当存在多个所述断点匹配字符时，多个断点匹配字符之间存在优先级，其中，按照优先级从高至低的顺序将多个断点匹配字符依次用于与所述文档数据片段进行匹配，直至获取相匹配的数据；以及

若与所述文档数据片段匹配成功，则调整相应的断点匹配字符对应的优先级。

7. 根据权利要求6所述的文档分割方法，其特征在于，若所述文档数据片段中不存在与所述预定义的断点匹配字符相匹配的数据，则扩大所述截取范围，以重新获取所述文档数据片段，并将重新获取的文档数据片段与所述断点匹配字符进行匹配，若仍不存在，则继续扩大所述截取范围和重新截取所述文档数据片段，直至查找到与所述断点匹配字符相匹配的数据，并根据该相匹配的数据所处的第二位置分割所述文档，以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。

8. 根据权利要求6所述的文档分割方法，其特征在于，在所述文档对应的字节数据中确定所述第一位置和截取所述数据片段；以及

获取所述断点匹配字符对应的字节数据，并在所述文档数据片段中进行匹配，以确定所述第二位置。

9. 根据权利要求8所述的文档分割方法，其特征在于，还包括：

获取所述文档使用的字符集，以用于对所述断点匹配字符进行转换；以及

利用转换后的断点匹配字符对所述文档数据片段进行匹配。

10. 根据权利要求6至9中任一项所述的文档分割方法，其特征在于，所述断点匹配字符包括标点符号。

## 文档分割系统和文档分割方法

### 技术领域

[0001] 本发明涉及文档处理技术领域,具体而言,涉及一种文档分割系统和一种文档分割方法。

### 背景技术

[0002] 在用户进行数字阅读时,经常会需要在文档的内容间进行跳转,比如从起始页跳转至文档内容的55%处。而根据文档内容或来源的不同,不同的文档往往采用不同的字符集进行显示。在不同的字符集中,每个字符所占字节数不尽相同,具体如下表所示:

[0003]

字符集	英文字节数	中文字节数
GB2312	1	2
GBK	1	2
GB18030	1	2
ISO-8859-1	1	1
UTF-8	1	3
UTF-16	4	4
UTF-16BE	2	2
UTE-16LE	2	2

[0004] 表1

[0005] 从表1可以看出,在某些字符集中,中文与英文或符号的字节数是不同的(如UTF-8),而一些符号和外文,字节数更是多样化。这样在进行文档跳转时,就不能方便的知道字节流的某一位置是否是某一字符的开始位置。当跳转到这一位置并显示其内容时,就有可能并不是一个完整字符的起始位置。

[0006] 现有的常用做法是,将字节数据定位到某一位置posA时,若posA的位置相对于当前显示位置靠前,则从起始位置开始遍历数据;若posA的位置相对于当前显示位置靠后,则从当前位置开始遍历数据。遍历数据时计算累加对应字符集中每个字符的长度,直到我们要定位到的位置,然后检测是否定位到了完整的字符处,进而处理posA的值;这样,如果posA的值稍大,就会出现耗时过长和内存消耗过大的问题。

[0007] 因此,需要一种新的文档分割技术,可以针对用户的跳转需求,快速准确地实现文档分割,避免分割处的字符不完整。

### 发明内容

[0008] 本发明正是基于上述问题,提出了一种新的文档分割技术,可以针对用户的跳转需求,快速准确地实现文档分割,避免分割处的字符不完整。

[0009] 有鉴于此,本发明提出了一种文档分割系统,包括:指令处理单元,用于根据接收到的跳转指令和预设的页面跳转规则,确定当前文档中的第一位置;数据截取单元,用于按

照预设的截取范围对所述第一位置附近的文档数据片段进行截取；数据匹配单元，用于将所述文档数据片段与预定义的断点匹配字符进行匹配；文档分割单元，用于在所述文档数据片段中存在与所述断点匹配字符相匹配的数据的情况下，根据该相匹配的数据所处的第二位置分割所述文档，以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。

[0010] 在该技术方案中，第一位置是根据现有技术规定的跳转规则获取的，但若直接根据第一位置进行文档分割，则可能造成分割处的字符不完整，影响分割效果。而通过将预设的断点匹配字符与第一位置附近的文档数据片段进行比较，由于断点匹配字符肯定是完整的字符，因而在根据第二位置分割后，使得断点匹配字符作为分割得到的后一个文档片段的起始端，则能够确保分割得到的多个文档片段在分割处均为完整的字符。当然，根据实际情况的不同，这里的第一位置与第二位置可以是相同的（即按照跳转规则进行计算后，得到的第一位置处的字符正好是断点匹配字符），也可以是不同的。

[0011] 根据本发明的又一方面，还提出了一种文档分割方法，包括：根据接收到的跳转指令和预设的页面跳转规则，确定当前文档中的第一位置；按照预设的截取范围对所述第一位置附近的文档数据片段进行截取；若所述文档数据片段中存在与预定义的断点匹配字符相匹配的数据，则根据该相匹配的数据所处的第二位置分割所述文档，以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。

[0012] 在该技术方案中，第一位置是根据现有技术规定的跳转规则获取的，但若直接根据第一位置进行文档分割，则可能造成分割处的字符不完整，影响分割效果。而通过将预设的断点匹配字符与第一位置附近的文档数据片段进行比较，由于断点匹配字符肯定是完整的字符，因而在根据第二位置分割后，使得断点匹配字符作为分割得到的后一个文档片段的起始端，则能够确保分割得到的多个文档片段在分割处均为完整的字符。当然，根据实际情况的不同，这里的第一位置与第二位置可以是相同的（即按照跳转规则进行计算后，得到的第一位置处的字符正好是断点匹配字符），也可以是不同的。

[0013] 通过以上技术方案，可以针对用户的跳转需求，快速准确地实现文档分割，避免分割处的字符不完整。

## 附图说明

[0014] 图1示出了根据本发明的实施例的文档分割系统的框图；

[0015] 图2示出了根据本发明的实施例的文档分割方法的流程图；

[0016] 图3和图4示出了根据本发明的实施例的使用断点匹配字符进行数据匹配的示意图；

[0017] 图5示出了根据本发明的实施例的分割文档的具体流程图。

## 具体实施方式

[0018] 为了能够更清楚地理解本发明的上述目的、特征和优点，下面结合附图和具体实施方式对本发明进行进一步的详细描述。需要说明的是，在不冲突的情况下，本申请的实施例及实施例中的特征可以相互组合。

[0019] 在下面的描述中阐述了很多具体细节以便于充分理解本发明，但是，本发明还可以采用其他不同于在此描述的其他方式来实施，因此，本发明并不限于下面公开的具体实

施例的限制。

[0020] 图1示出了根据本发明的实施例的文档分割系统的框图。

[0021] 如图1所示,根据本发明的实施例的文档分割系统100,包括:指令处理单元102,用于根据接收到的跳转指令和预设的页面跳转规则,确定当前文档中的第一位置;数据截取单元104,用于按照预设的截取范围对所述第一位置附近的文档数据片段进行截取;数据匹配单元106,用于将所述文档数据片段与预定义的断点匹配字符进行匹配;文档分割单元108,用于在所述文档数据片段中存在与所述断点匹配字符相匹配的数据的情况下,根据该相匹配的数据所处的第二位置分割所述文档,以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。

[0022] 在该技术方案中,第一位置是根据现有技术规定的跳转规则获取的,但若直接根据第一位置进行文档分割,则可能造成分割处的字符不完整,影响分割效果。而通过将预设的断点匹配字符与第一位置附近的文档数据片段进行比较,由于断点匹配字符肯定是完整的字符,因而在根据第二位置分割后,使得断点匹配字符作为分割得到的后一个文档片段的起始端,则能够确保分割得到的多个文档片段在分割处均为完整的字符。当然,根据实际情况的不同,这里的第一位置与第二位置可以是相同的(即按照跳转规则进行计算后,得到的第一位置处的字符正好是断点匹配字符),也可以是不同的。

[0023] 在上述技术方案中,优选地,所述数据截取单元104还用于:在所述文档数据片段中不存在与所述预定义的断点匹配字符相匹配的数据的情况下,则扩大所述截取范围,以重新获取所述文档数据片段;其中,所述数据匹配单元106将重新获取的文档数据片段与所述断点匹配字符进行匹配,若仍不存在,则继续由所述数据截取单元104扩大所述截取范围和重新截取所述文档数据片段,直至查找到与所述断点匹配字符相匹配的数据,并由所述文档分割单元108根据该相匹配的数据所处的第二位置分割所述文档,以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。

[0024] 在该技术方案中,通过在没有查找到与断点匹配字符相匹配的数据时,扩大截取范围,从而得到包含更多数据的文档数据片段,提高查找到与断点匹配字符相匹配的数据的概率。具体地,比如原本的截取范围是由第一位置向后截取,则在对截取范围进行扩大时,可以向前截取,也可以继续向后截取;对于原本是由第一位置向前或向两侧截取的情况,与上述情况类似,此处不再赘述。

[0025] 在上述技术方案中,优选地,所述指令处理单元102在所述文档对应的字节数据中确定所述第一位置;所述数据截取单元104在所述文档对应的字节数据中截取所述数据片段;以及所述数据匹配单元106获取所述断点匹配字符对应的字节数据,并在所述文档数据片段中进行匹配,以确定所述第二位置。

[0026] 在该技术方案中,为了方便对文档的分割和对断点匹配字符的查找,可以将所有的字符都转换为对应的字节数据,从而有利于提高匹配和分割的效率。

[0027] 在上述技术方案中,优选地,还包括:字符转换单元110,用于获取所述文档使用的字符集,以用于对所述断点匹配字符进行转换;其中,所述数据匹配单元106利用转换后的断点匹配字符对所述文档数据片段进行匹配。

[0028] 在该技术方案中,由于不同的文档可能使用不同的字符集,而对于不同字符集中,相同字符对应的字节数是不同的,因此,为了准确地通过断点匹配字符进行匹配,需要确定

待分割的文档使用的字符集，并将断点匹配字符按照文档的字符集进行转换，以用于确定文档数据片段中是否存在对应的匹配数据。

[0029] 在上述技术方案中，优选地，所述断点匹配字符包括标点符号。

[0030] 在该技术方案中，由于每个文档中都必然存在标点符号，而标点符号对应的字节数据是必然可以预先肯定的，因而可以被作为断点匹配字符，以实现对文档的准确分割。

[0031] 在上述技术方案中，优选地，当存在多个所述断点匹配字符时，多个断点匹配字符之间存在优先级，其中，所述数据匹配单元106按照优先级从高至低的顺序将多个断点匹配字符依次用于与所述文档数据片段进行匹配，直至获取相匹配的数据；以及所述文档分割系统100还包括：优先级调整单元112，用于在与所述文档数据片段匹配成功的情况下，调整相应的断点匹配字符对应的优先级。

[0032] 在该技术方案中，断点匹配字符可能有很多，在对文档数据片段进行匹配时，每次使用其中的一个断点匹配字符，但并不是每个断点匹配字符都会被匹配到，当一个断点匹配字符没有被匹配到时，将使用其他的断点匹配字符继续进行匹配操作。并且对于某些断点匹配字符而言，较之其他的断点匹配字符而言，更有可能出现在文档中，因此，为了节省文档的分割时间，可以直接使用这些更有可能出现在文档中的断点匹配字符。具体地，为了确定每个断点匹配字符可能出现的概率大小，可以根据每次匹配操作的成功率，为成功率更高的断点匹配字符设置更高的优先级，以使其优先被用于字符匹配。

[0033] 图2示出了根据本发明的实施例的文档分割方法的流程图。

[0034] 如图2所示，根据本发明的实施例的文档分割方法，包括：步骤202，根据接收到的跳转指令和预设的页面跳转规则，确定当前文档中的第一位置；步骤204，按照预设的截取范围对所述第一位置附近的文档数据片段进行截取；步骤206，判断文档数据片段中是否存在与预定义的断点匹配字符相匹配的数据；步骤208，若存在，则根据该相匹配的数据所处的第二位置分割所述文档，以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。

[0035] 在该技术方案中，第一位置是根据现有技术规定的跳转规则获取的，但若直接根据第一位置进行文档分割，则可能造成分割处的字符不完整，影响分割效果。而通过将预设的断点匹配字符与第一位置附近的文档数据片段进行比较，由于断点匹配字符肯定是完整的字符，因而在根据第二位置分割后，使得断点匹配字符作为分割得到的后一个文档片段的起始端，则能够确保分割得到的多个文档片段在分割处均为完整的字符。当然，根据实际情况的不同，这里的第一位置与第二位置可以是相同的（即按照跳转规则进行计算后，得到的第一位置处的字符正好是断点匹配字符），也可以是不同的。

[0036] 在上述技术方案中，优选地，还包括：步骤210，若所述文档数据片段中不存在与所述预定义的断点匹配字符相匹配的数据，则扩大所述截取范围，以重新获取所述文档数据片段，并返回步骤206，将重新获取的文档数据片段与所述断点匹配字符进行匹配，若仍不存在，则进入步骤210中，并继续扩大所述截取范围和重新截取所述文档数据片段，直至查找到与所述断点匹配字符相匹配的数据，则进入步骤208，并根据该相匹配的数据所处的第二位置分割所述文档，以使所述相匹配的数据作为分割得到的后一个文档片段的起始端。

[0037] 在该技术方案中，通过在没有查找到与断点匹配字符相匹配的数据时，扩大截取范围，从而得到包含更多数据的文档数据片段，提高查找到与断点匹配字符相匹配的数据

的概率。具体地,比如原本的截取范围是由第一位置向后截取,则在对截取范围进行扩大时,可以向前截取,也可以继续向后截取;对于原本是由第一位置向前或向两侧截取的情况,与上述情况类似,此处不再赘述。

[0038] 在上述技术方案中,优选地,在所述文档对应的字节数据中确定所述第一位置和截取所述数据片段;以及获取所述断点匹配字符对应的字节数据,并在所述文档数据片段中进行匹配,以确定所述第二位置。

[0039] 在该技术方案中,为了方便对文档的分割和对断点匹配字符的查找,可以将所有的字符都转换为对应的字节数据,从而有利于提高匹配和分割的效率。

[0040] 在上述技术方案中,优选地,还包括:获取所述文档使用的字符集,以用于对所述断点匹配字符进行转换;以及利用转换后的断点匹配字符对所述文档数据片段进行匹配。

[0041] 在该技术方案中,由于不同的文档可能使用不同的字符集,而对于不同字符集中,相同字符对应的字节数是不同的,因此,为了准确地通过断点匹配字符进行匹配,需要确定待分割的文档使用的字符集,并将断点匹配字符按照文档的字符集进行转换,以用于确定文档数据片段中是否存在对应的匹配数据。

[0042] 在上述技术方案中,优选地,所述断点匹配字符包括标点符号。

[0043] 在该技术方案中,由于每个文档中都必然存在标点符号,而标点符号对应的字节数据是必然可以预先肯定的,因而可以被作为断点匹配字符,以实现对文档的准确分割。

[0044] 在上述技术方案中,优选地,还包括:当存在多个所述断点匹配字符时,多个断点匹配字符之间存在优先级,其中,按照优先级从高至低的顺序将多个断点匹配字符依次用于与所述文档数据片段进行匹配,直至获取相匹配的数据;以及若与所述文档数据片段匹配成功,则调整相应的断点匹配字符对应的优先级。

[0045] 在该技术方案中,断点匹配字符可能有很多,在对文档数据片段进行匹配时,每次使用其中的一个断点匹配字符,但并不是每个断点匹配字符都会被匹配到,当一个断点匹配字符没有被匹配到时,将使用其他的断点匹配字符继续进行匹配操作。并且对于某些断点匹配字符而言,较之其他的断点匹配字符而言,更有可能出现在文档中,因此,为了节省文档的分割时间,可以直接使用这些更有可能出现在文档中的断点匹配字符。具体地,为了确定每个断点匹配字符可能出现的概率大小,可以根据每次匹配操作的成功率,为成功率更高的断点匹配字符设置更高的优先级,以使其优先被用于字符匹配。

[0046] 下面通过一具体实施例,并结合图3至图5对本发明的技术方案进行详细说明。

[0047] 实例:快速截断TXT文件的字节流,并假定该字节流采用的字符集为GBK。

[0048] 对该字节流进行分割的具体流程如图5所示:

[0049] 步骤502,定位至posA。具体地,是指根据现有技术中规定的文档跳转规则,并按照用户发出的跳转指令(如需要跳转至文档的55%处),确定其在文档中的一个分割点posA。但需要说明的是,该posA可能导致其对应的字符被分割。

[0050] 具体地,比如待分割文档的源数据为:

[0051] “由上可知,同一字符集下,中文与英文(如ABC)的字节数可能是不同的,这样就不能方便的知道字节流的某一位置是否是某一字符的开始位置。那么当跳转到某一位置并显示其内容时,势必遇到字节流截取问题,我们要保证截取点在一个完整的字符处。”

[0052] 该源数据对应的字节数据为:

[0053] “-45-55-55-49-65-55-42-86-93-84-51-84-46-69-41-42-73-5-68-81-49-62-93-84-42-48-50-60-45-21-45-94-50-6040-56-2565666741-75-60-41-42-67-38-54-3-65-55-60-36-54-57-78-69-51-84-75-60-93-84-43-30-47-7-66-51-78-69-60-36-73-67-79-29-75-60-42-86-75-64-41-42-67-38-63-9-75-60-60-77-46-69-50-69-42-61-54-57-73-15-54-57-60-77-46-69-41-42-73-5-75-60-65-86-54-68-50-69-42-61-95-93-60-57-61-76-75-79-52-8-41-86-75-67-60-77-46-69-50-69-42-61-78-94-49-44-54-66-58-28-60-38-56-35-54-79-93-84-54-58-79-40-45-10-75-67-41-42-67-38-63-9-67-40-56-95-50-54-52-30-93-84-50-46-61-57-46-86-79-93-42-92-67-40-56-95-75-29-44-38-46-69-72-10-51-22-43-5-75-60-41-42-73-5-76-90-95-93”

[0054] 当按照上述文档跳转规则进行跳转时,假定确定的posA=45,即以上述字节数据的第一个字节“-45”为第1个,顺次向后数到第45个,即“-42”(在上述字节数据中字体加粗处理)。但该“-42”正是源数据中的“字”(在上述源数据中字体加粗处理)的字节数据的一半(另一半为“-42”之后的“-67”,由“-42-67”构成了“字”),因而若从此处分割文档,将导致字符不完整。

[0055] 步骤504,读取posA附近的一段数据至字节数组ArrayA。

[0056] 具体地,比如读取“-42-67-38-54-3-65-55-60-36-54-57-78-69-51-84-75-60-93-84-43-30-47-7-66-51-78-69-60-36-73-67-79-29-75-60-42-86-75-64-41-42-67-38-63-9-75-60-60-77-46”。这里是从posA开始向后读取的一段数据,当然用户可以根据实际情况或使用习惯,选择从posA开始向前读取一段数据,或是同时向posA的前后两侧读取数据,并且对于每侧读取的数据的数量也可以由用户自行确定。

[0057] 步骤506,获取断点匹配字符。

[0058] 这里的断点匹配字符是由用户事先设定的,具体可以是一些用户确定有较大可能出现在文档中的完整字符,从而确保当从这些断点匹配字符处分割文档时,实现对文档的完整分割。具体地,断点匹配字符可以为一些文档中常见的字符,如:句号、逗号、分号、回车换行等。

[0059] 本例列三个断点匹配字符举例说明:“#”、“,”、“。”,其中:

[0060] List<String>matchStrings=newArrayList<String>;

[0061] matchStrings.add("#");

[0062] matchStrings.add(",");

[0063] matchStrings.add(".").

[0064] 步骤508,转换为对应编码的字节数组。由于每个文档可能采用不同的字符集,而每个字符集中的字节数不尽相同,因此需要确定文档采用的字符集,并对断点匹配字符进行转换。比如这里的文档采用的是GBK字符集,则“#”、“,”、“。”对应的字节数据分别为“35”、“-93-84”、“-95-93”。

[0065] 步骤510,判断读取在ArrayA中的字节数据片段中是否存在上述断点匹配字符。

[0066] 具体地,如图3所示,取“#”对应的“35”:

[0067] Byte[]matchBytes=35;

[0068] 然后将ArrayA中的字节依次取出后与“35”进行匹配。此处最终的匹配结果为失败。

- [0069] 则返回步骤506，重新选取另一个断点匹配字符，继续进行匹配。
- [0070] 假定第二次取出的是“，”，则如图4所示，取“，”对应的“-93 -84”：
- [0071] Byte[]matchBytes=-93-84;
- [0072] 然后将ArrayA中的字节依次取出后与“-93-84”进行匹配。此处最终的匹配结果为成功。
- [0073] 步骤512，确定匹配字符在读取数据中的位置。具体地，与“-93-84”匹配成功的字节在读取的字节数据片段中所处的位置为18(即以“-42”为第1个字节，则“-93”为第18个字节)。
- [0074] 步骤514，将posA改变为posB，其中， $posB = posA + 18 = 45 + 18 = 63$ ，则从posB=63处进行分割时，分割后可以确保文档的字符完整性。
- [0075] 此外，当某个断点匹配字符与读取数据匹配成功后，若存在多个断点匹配字符，则可以对这些断点匹配字符的优先级进行调整，使得匹配成功的断点匹配字符更优先地被用于与读取数据进行匹配。具体地，比如此处可以将“，”与“#”互换位置，即“，”的匹配优先级提高，则下次分割字节数据时，将优先使用“，”进行匹配，以提高首次匹配成功的概率，节省匹配时间。当然，此处匹配符优先级算法包含但不仅限于此算法。
- [0076] 以上结合附图详细说明了本发明的技术方案，考虑到相关技术中，对于文档分割的效率低、对内存消耗大，因此，本发明提出了一种文档分割系统和一种文档分割方法，可以实现以下优点：
- [0077] 1、提高了分割的速度；
- [0078] 2、优化了内存的使用；
- [0079] 3、保证了字节流分割的正确性，保证截取到完整的字符处；
- [0080] 4、动态调整匹配成功的匹配符的权重，提高首次匹配成功的概率。
- [0081] 以上所述仅为本发明的优选实施例而已，并不用于限制本发明，对于本领域的技术人员来说，本发明可以有各种更改和变化。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。



图1

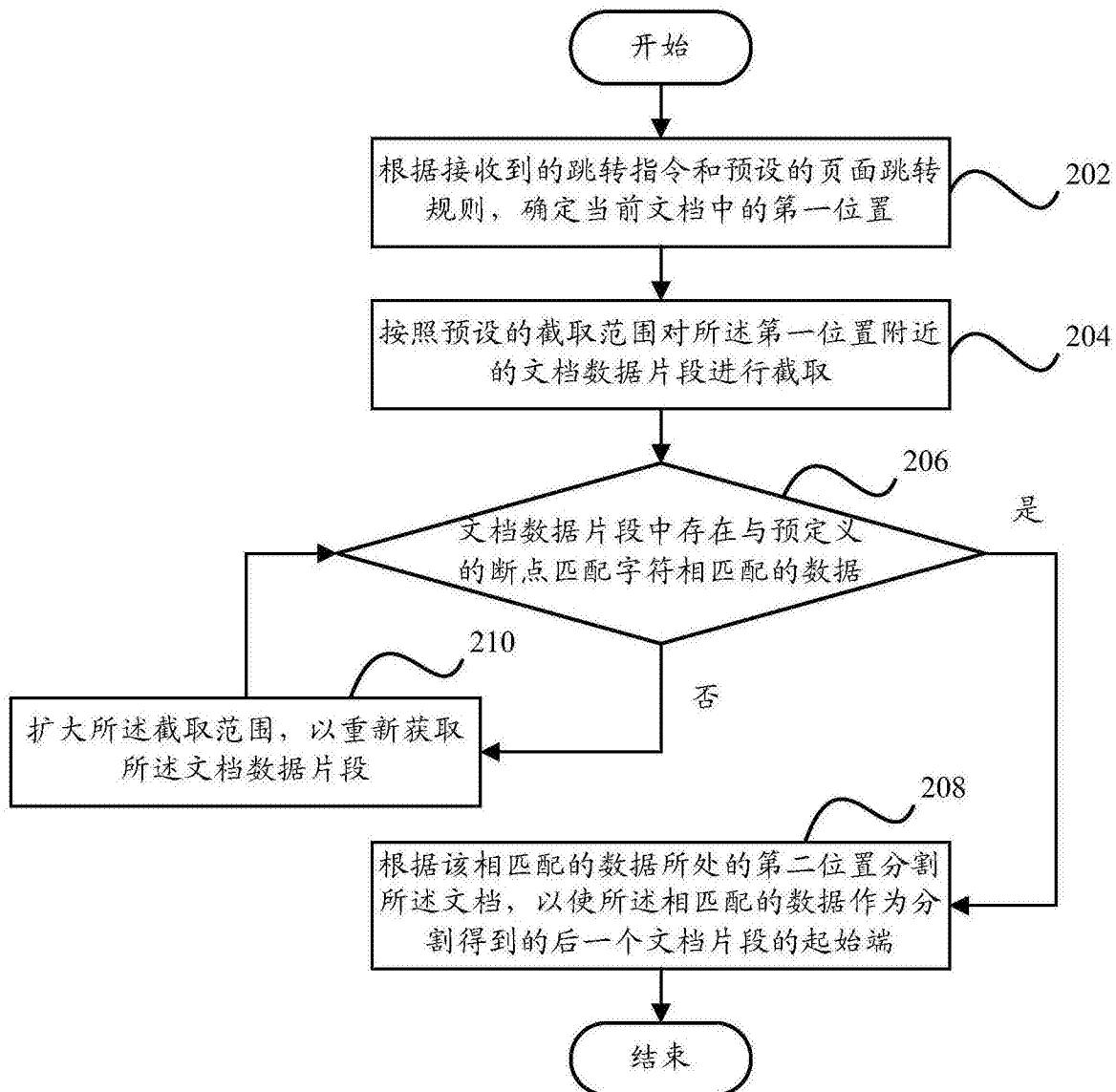


图2

arrayAToMatch:

```
-33 -67 -38 -54 -3 -65 -55 -60 -36 -54 -57 -78 -69 -51 -84 -75 -60 -93 -84 -43 .....
```

matchBytes:

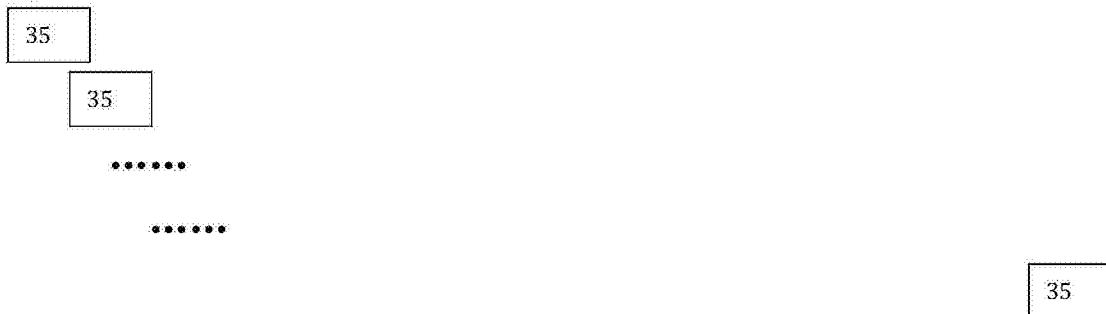


图3

arrayAToMatch:

```
-33 -67 -38 -54 -3 -65 -55 -60 -36 -54 -57 -78 -69 -51 -84 -75 -60 -93 -84 -43 .....
```

matchBytes:



图4

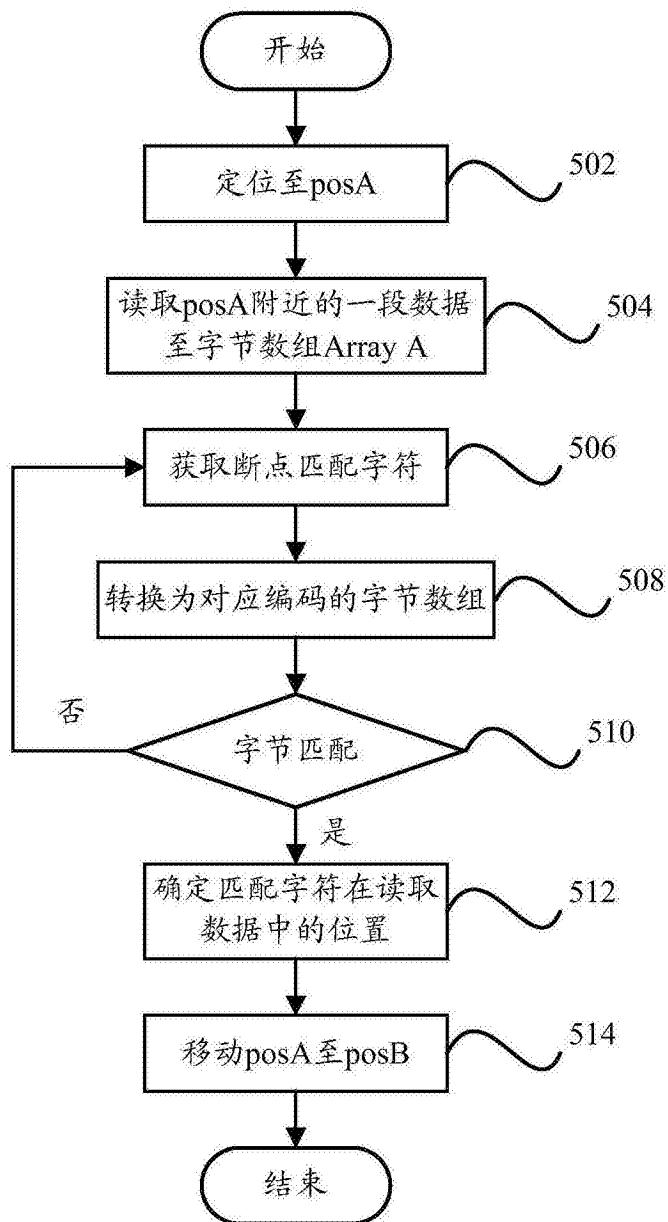


图5