计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190900116

多级字典存储的未知文本协议候选关键词链式合并方法

陈庆超[1]　王　韬[1]　尹世庄[1]　冯文博[2]
1 陆军工程大学装备模拟训练中心　石家庄 050003
2 陆军工程大学指挥控制工程学院　南京 210007
(cqc62808@163.com)

摘　要　关键词提取是进行未知网络协议逆向的关键步骤。鉴于现有的关键词提取方法存在精确度不高、需要较多先验知识、操作繁琐等问题,提出了一种基于位置信息的关键词自动化提取算法。首先,通过 Trigram 分词获取候选关键词,附加上位置信息后,将其组织成多级字典;在此基础上,根据位置信息将传统的对候选关键词进行树状合并改进为对其进行链式合并,以获得更精确的最长候选关键词。实验结果表明,当设置频繁度阈值为 0.6 时,该方法即可以准确提取出文本协议的关键词。同时,分析了频繁度的设置对实验效果的影响,并讨论了基于频繁序列对关键词进行挖掘的相关算法的局限性。

关键词:位置信息;关键词提取;未知文本协议;Trigram;链式;多级字典
中图法分类号　TP393

Chain Merging Method for Unknown Text Protocol Candidate Keyword Stored in Multi-level Dictionary

CHEN Qing-chao[1], WANG Tao[1], YIN Shi-zhuang[1] and FENG Wen-bo[2]
1 Equipment Simulation Training Center, Army Engineering University, Shijiazhuang 050003, China
2 College of Command and Control Engineering, Army Engineering University, Nanjing 210007, China

Abstract　Keyword extraction is a key step in the reverse engineering of unknown network protocols. The existing keyword extraction methods have some problems, such as low accuracy, complex operation and more prior knowledge is required. Therefore, an automatic keyword extraction algorithm based on location information is proposed. First, the candidate keywords are obtained by Trigram word segmentation. After adding the location information, these keywords are organized into a multi-level dictionary. On this basis, the traditional tree merging of candidate keywords is improved to chain merging according to the location information, so as to obtain more precise and the longest candidate keywords. The experimental results show that, when the frequency threshold is set to 0.6, this method can accurately extract the keywords of text protocol. At the same time, the influence of frequency setting on experimental result is analyzed, and the limitations of related algorithms for keyword mining based on frequent sequences are also discussed.

Keywords　Location information, Keyword extraction, Unknown text protocol, Trigram, Chain, Multi-level dictionary

1　引言

未知网络协议逆向,是指在对未知协议没有先验知识的前提下,对协议的格式、状态机、语义等进行推断的研究[1]。协议格式的获取是对未知网络协议逆向的首要步骤,目前的协议格式推断方法主要分为序列对比[2-5]和关键词提取[6-9]两大类。基于序列对比的协议格式推断算法需要获取大量的相同协议格式的数据,时间复杂度和空间复杂度高,且容错能力不强。基于关键词提取的协议格式推断算法虽然具有一定的容错能力,但是容易忽略一些出现不频繁的格式信息。本文研究基于关键词提取的协议格式推断算法,相比二进制协议,文本协议具有更高的复用性和可读性,开发的时间成本也较低,应用更加广泛,因此本文方法选择未知文本协议作为研究对象。

目前,对于未知文本协议的关键词提取已经取得了很多成果。Krueger 利用自然语言处理中使用的 N-gram 算法[10]对报文序列进行分词,获取候选关键词[11],进而提出了基于统计的格式推断算法 PRISMA。Zhang 等提出了基于投票专家系统的关键词提取算法 ProWord,其利用块内信息熵和字节边界信息熵合并关键词[12]。Luo 等提出了基于位置信息

到稿日期:2019-09-17　返修日期:2019-11-21　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家重点研发计划(2017YFB0802900);江苏省自然科学基金(BK20161469)
This work was supported by the National Basic Research Program of China(2017YFB0802900) and Natural Science Foundation of Jiangsu Province, China (BK20161469).
通信作者:王韬(a13592247640@foxmail.com)

的 AutoReEngine 算法，并在算法中引入了换行符等先验知识，利用频繁序列进行挖掘，但是只对报文的首尾、段首尾的位置进行统计，并且根据候选关键词出现位置的方差排除部分位置变化较大的候选关键词[13]。Hong 等提出了基于扩展前缀树的协议格式推断算法，该算法利用 N-gram 分词获取候选关键词，然后结合点间互信息对候选关键词进行合并，但该方法存在较高局限性[14]。Hou 等提出了基于位置的自动化网络协议逆向方法，对二进制协议进行关键词提取；但是关键词是从一字节开始提取，且关键词合并是从长度为 K 字节向 $K+1$ 字节合并，算法的空间复杂度和时间复杂度都较高，关键词合并方法存在一定的局限性[15]。Li 等提出了一种基于 PrefixSpan[16] 的 PositionSpan 算法，按照偏移依次挖掘报文不同位置上的 1-位置频繁字节项，然后按照先后次序从 1-位置频繁字节项进行映射得到频繁字符串集合；但是算法中获取的是相对于报文首部固定位置的频繁序列，不能获取处于不确定长度的用户数据后的关键词[17]。

针对未知文本协议关键词提取算法中的时间复杂度和空间复杂度高等问题，本文提出了一种基于位置信息的未知文本协议关键词提取算法。该算法首先使用 Trigram 算法获取长度为 3 的候选关键词的集合，并给每一个关键词附加位置信息；然后根据频繁度对候选关键词进行筛选，排除出现频率不高的候选关键词；最后根据关键词出现的相对位置对候选关键词进行合并，从而获得更加精准的候选关键词集合。

2 基于位置信息的未知文本协议关键词的提取

现有的关键词提取算法的主要思想是频繁序列挖掘。一般地，基于 Apriori[18] 性质，即频繁序列的子集还是频繁序列，通过将短频繁序列合并成长频繁序列完成频繁序列的挖掘，继而实现关键词的提取。使用频繁序列挖掘算法提取文本协议关键词的主要问题集中在关键词的合并上：1）对候选关键词的选取从 1 字节长度开始，而实际上关键词的长度往往不小于 3 字节长度，因此存在从 1 字节向 2 字节再向 3 字节长度的候选关键词合并，从而占用大量时间和空间的问题；2）对候选关键词的合并一般是从 K 长度候选关键词的集合合并至 $K+1$ 长度的候选关键词的集合，期间进行了重复的计算，对所有长度的候选关键词的保存也占用了一定的存储空间。本文的主要创新在于对这两个问题进行了改进。

2.1 方案的总体设计

本文采取 Trigram 算法来挖掘所有长度为 3 字节的序列，进而筛选出频繁度高于一定阈值的序列作为初始候选关键词。

传统的关键词合并是由长度为 K 字节的候选关键词合并为长度为 $K+1$ 的候选关键词，如将 abcde 和 bcdef 合并，之后判断 abcdef 的频繁度，因此需要存储所有长度的候选关键词且保存了大量中间状态的候选关键词。针对这个问题，本文为候选关键词添加了位置属性，关键词之间可以直接根据位置上的相对差进行合并，循环查找关键词集合，可以直接生成最长的候选关键词，不需要保存中间状态的候选关键词。传统合并算法的整体结构是树状结构，如图 1 所示；改进后的合并算法是链状结构，如图 2 所示，减少了时间和空间的开销。

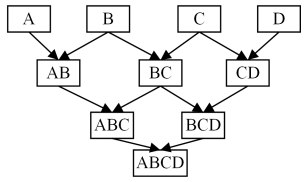

图 1 传统树状合并结构

Fig. 1 Traditional tree merging structure

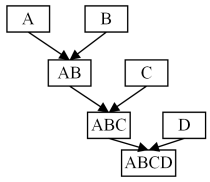

图 2 链式合并结构

Fig. 2 Chain merging structure

基于位置信息的未知文本协议关键词提取算法的主体流程为：首先使用 Trigram 算法获取长度为 3 的候选关键词集合，并给每一个关键词添加位置信息；然后根据频繁度对候选关键词进行筛选，排除出现频率不高的候选关键词；最后根据关键词出现的相对位置对候选关键词进行合并，从而获得更加精确的候选关键词集合。基于位置信息的未知文本协议关键词提取算法可以一次性提取出所有长度的超过频繁度阈值的关键词。

2.2 基于 Trigram 算法的分词

Trigram 算法即设置分词长度为 3 的 N-gram 算法。原始报文集合可以表示为 $DATASET=\{M_1,M_2,\cdots,M_i,\cdots,M_n\}$，其中 M_i 为原始报文集合中的第 i 个报文，$M_i=W_1W_2W_3\cdots W_j\cdots W_s$，$W_j$ 表示报文 M_i 的第 j 个字节。对原始报文集合中的每一个报文进行 Trigram 分词处理，将得到的候选序列集合组合成候选关键词属性集合 $KeySet=\{set_1,set_2,\cdots,set_{ii}\cdots set_{jj}\}$，其中 set_{ii} 为候选关键词属性集合 $KeySet$ 中的第 ii 个候选关键词属性。本文方法对候选关键词添加位置属性，位置属性包括关键词出现的报文的序号和关键词出现在报文中的位置，故候选关键词属性可表示为 $set_{ii}=\langle key:\{num_1:pos_1;\ num_2:pos_2;\cdots;\ num_i:pos_i;\ \cdots;num_j:pos_j\}\rangle$，其中 key 表示候选关键词的序列值，$num_i:pos_i$ 表示该候选关键词序列值出现在第 num_i 个报文的 pos_i 位置。因为是以字典形式存储候选关键词序列的位置，所以一个报文只能记录候选关键词的一次位置，故在统计关键词时，如果一个字符串在一个报文中出现多次，则只对其进行一次统计，并记录其第一次出现的位置。根据统计特性，少量数据字段与关键词字段重复，并不影响统计的整体效果。记录候选关键词出现的所有位置的实验结果表明，只记录候选关键词第一次出现的位置对关键词的提取影响不大。

2.3 基于位置信息的候选关键词合并

本文基于候选关键词的位置关系对其进行合并，得到不同长度的协议关键词。下面给出合并的具体步骤。

从候选关键词属性集合中选出一个元素 P，遍历候选关键词属性集合，直至遍历到元素 Q。设 $P=\langle key_1:\{num_1:pos_1;\cdots;num_i:pos_i;\cdots;num_k:pos_k\}\rangle$，$Q=\langle key_2:\{nu_1:po_1;\cdots;nu_i:po_i;\cdots;nu_k:po_k\}\rangle$，则：

1）判断 key_1 的后两个字节是否等于 key_2 的前两个字节，或者 key_1 的前两个字节是否等于 key_2 的后两个字节（假设前一种情况成立）。

2）如果 1）成立，则判断报文集合中 P 和 Q 的关键词序列共同出现的报文的数量是否大于阈值 $Tfreq$，即 $len(P[key_1].keys()\ \&\ Q[key_2].keys())>Tfreq$ 是否成立。

3）如果 2）成立，则判断 P 和 Q 的关键词序列共同出现的

报文中满足两个关键词序列相邻的数量是否大于 $Tfreq$。

4)如果 3)成立,则将两个候选关键词属性合并,组成新的候选关键词属性,并用其替代 P,同时删除 Q 和原 P,继续遍历候选关键词属性集合,直至没有序列满足条件。

该过程的伪代码描述如算法 1 所示。

算法 1 基于位置信息的关键词合并算法

```
输入:候选关键词属性集合 count_dict
输出:关键词集合 ITEM
BEGIN
1. list1=list(count_dict.keys)
2. listdict=dict()
3. for thing in list1:
4.     listdict[thing]=0 #初始标记
5.  for item in list1:
6.     if listdict[item] ==1:
7.        continue #判断是否标记
8.     for item1 in list1:
9.        if listdict[item1] ==1
10.          continue
11.       if item[-2:0] ==item1[0:2]: # 另一种情况的运行方式
          相同,不再赘述
12.          summ=0 #计数,与阈值进行比较
13.          keylist=两个关键词共同出现的报文序号集合
14.          for key in ( keylist):
15.             if (报文中两个关键词相邻):
16.                summ=sum+1
17.                将 item 的相同报文上的位置信息赋值给合并后的
                   item+item[2:]
18.          if sum>Tfreq:
19.             标记 listdict 中被合并的候选关键词
20.             item=item+item1[2:]
21.          else:
22.                将步骤 17 产生的新的候选关键词 item+item[2:] 删除
23.    ITEM.append(item)
END
```

相比传统的由两个长度为 K 的候选关键词合并为长度为 $K+1$ 的候选关键词,本文的 $K+1$ 长度的候选关键词是由长度为 K 的候选关键词和长度为 3 的候选关键词合并而成,将合并过程组织成了整体上的链式结构,相对缩短了运算时间。

3 实验结果及分析

本文在配置为 2.6 GHz 的 CPU,8 GB 内存,操作系统为 Windows 10 的 PC 上基于 Python 语言实现了本文方法。AutoReEngine 算法引入了换行符等先验知识,关键词提取的准确率高,具有一定的代表性,因此将其与本文方法进行实验比较。本文的实验对象为 HTTP 协议,主要考虑两个方面:1)HTTP 协议为较常见的协议,数据的获取以及后续实验效果的验证也相对容易;2)本文方法并未针对协议特点进行设计,具有一定的普适性,算法的有效性验证与选择何种协议无关,而与选择协议的数据分布好坏有关。目前,未知文本协议结构推断研究中的常用研究对象也包括 HTTP 协议。

设置不同的阈值 $Tfreq$,针对 HTTP 协议进行实验。实验数据的总体情况如表 1 所列。

表 1 实验数据的总体情况

Table 1 Overview of experimental data

Type	quantity
HTTP 协议	2070
GET	1583
OPTION	307
PROFIND	80
HTTP	100

HTTP 报文的总量为 2070 条,其中以“GET”开头的有 1583 条,以“OPTION”开头的有 307 条,以“PROFIND”开头的有 80 条,以 HTTP 开头的有 100 条。

HTTP 协议数据中的主流协议数据类型“GET”的占比约为 0.76,因此分别设置阈值为 0.6,0.7 和 0.8,针对协议数据进行实验,得到的结果如表 2 所列,其中“_”代表空格。

表 2 不同阈值下本文算法的关键词提取结果

Table 2 Keyword extraction results of proposed algorithm under different thresholds

number	$Tfreq=0.6$	$Tfreq=0.7$	$Tfreq=0.8$
1	GET_/	GET_/	_HTTP/1.1\r\n
2	_HTTP/1.1\r\n	_HTTP/1.1\r\n	\r\nUser-Agent:
3	\r\nUser-Agent:	\r\nUser-Agent:	\r\n\r\n
4	\r\n\r\n	\r\n\r\n	\r\nHost:
5	\r\nHost:	\r\nHost:	t:_
6	\r\nAcce	t:_	:_c
7	0\r\n	:_c	
8	t:_		
9	:_c		

设阈值为 0.7,使用 AutoReEngine 算法对 HTTP 协议进行关键词提取,结果如表 3 所列。

表 3 AutoReEngine 算法的关键词提取结果

Table 3 Keyword extraction results of AutoReEngine

number	$Tfreq=0.7$
1	GET_/
2	_HTTP/1.1
3	User-Agent:
4	Host:

1)从表 2 可以看出,当设置合适的阈值时,得到的关键词和实际关键词基本上是符合的,其中提取的关键词总是附带“_”和“\r\n”,这是因为文本协议的解析是通过匹配分隔符来划分区域以进行解析的,所以关键词总是与分隔符相连。但是,因为某些关键词之间存在相同的序列片段,而这些关键词的出现频率不高,所以设置较高的阈值时,提取出了某些关键词的片段,如“t:_”和“:_c”。同时,由表 2 可得根据频繁序列获取关键词的普遍问题,即设置较高的阈值时,会导致某些出现频率不高的关键词被忽略。例如,设置阈值为 0.6 时,“OPTION”等被忽略;设置阈值为 0.8 时,“GET”等被忽略。而当设置较低的阈值时,又会出现用户数据和关键词数据之间的混杂。因此,根据频繁序列挖掘关键词只能挖掘出一类主流类型的关键词。阈值的设定决定了根据频繁序列挖掘关键词的效果。

2)对比表 2 和表 3 可知,相比 AutoReEngine 算法,本文方法提取的关键词更加丰富,使用 AutoReEngine 算法获得

的关键词的结果也在一定程度上验证了本文方法的有效性。虽然 AutoReEngine 算法中引入了换行符等先验知识，可以将本文方法获得的候选关键词“:_c”和“t:_”剔除，但是对于未知文本协议，分隔符存在较大的不确定性；并且 AutoReEngine 算法只能提取位置相对于报文首尾、段首尾大致固定的关键词，这样虽然提升了准确率，但是限制了对位置变化较大的关键词的提取。

3）观察表 2 可知，当设置阈值 $Tfreq$ 为 0.6 时，算法挖掘出\r\nAcce。对数据进行实验分析，发现当设置阈值为 0.5 时，可以准确挖掘出\r\nAccept:。关键词的部分片段和关键词的频繁度有差异，导致将关键词的部分片段误认为关键词本身，这也是基于频繁序列挖掘关键词需要解决的问题之一。

本文算法沿用 N-gram 算法，采用了空间换时间及精确率的策略，为了避免重复扫描原始数据集，选择给候选关键词附加上位置信息。构造初始候选关键词集合时，本文算法只需要对原始数据进行单次扫描，对数据进行重组，空间复杂度为 O(N)，其中 N 为原始数据集的大小。后续候选关键词合并过程若采用替换策略，则最终结果为最长的候选关键词；若采用添加策略，即保留原始候选关键词，同时添加合并产生的候选关键词，则最终输出的是所有不同长度的频繁序列的集合。本文算法具有与其他基于频繁序列推断未知协议格式的方法一样的局限性，即默认了关键词一定是频繁序列，实验数据的好坏很大程度上决定了实验效果。

结束语 本文提出了一种基于位置信息的未知文本协议关键词提取算法，通过 Trigram 分词，结合位置信息获取协议关键词，同时从初始候选关键词的提取和合并方式方面对传统的候选关键词提取算法进行了改进。与 AutoReEngine 算法的对比实验结果证明了本文算法的有效性。下一步，我们将循环提取关键词以提取出现频率低的关键词，通过序列对比完成对文本协议分隔符的提取，同时将本文方法应用到未知协议比特流的频繁序列挖掘中。

参 考 文 献

[1] DUCHENE J, LE GUERNIC C, ALATA E, et al. State of the art of network protocol reverse engineering tools[J]. Journal of Computer Virology and Hacking Techniques, 2018, 14(1): 53-68.

[2] Beddoe M A. Network protocol analysis using bioinformatics algorithms[OL]. http://www.4tphi.net/~awalters/PI/pi.pdf.

[3] SIJA B D, GOO Y H, SHIM K S, et al. A survey of automatic protocol reverse engineering approaches, methods, and tools on the inputs and outputs view[J]. Security and Communication Networks, 2018, 2018: 1-17.

[4] CUI W, KANNAN J, WANG H J. Discoverer: Automatic Protocol Reverse Engineering from Network Traces[C]// USENIX Security Symposium. 2007: 1-14.

[5] PAN F, HONG Z, DU Y Y, et al. Recursive Clustering Based Method for Message Structure Extraction[J]. Journal of Sichuan University (Engineering Science Edition), 2012, 44(6): 137-142.

[6] BISWAS S K, BORDOLOI M, SHREYA J. A graph based keyword extraction model using collective node weight [J]. Expert Systems with Applications, 2018, 97: 51-59.

[7] KLEBER S, MAILE L, KARGL F. Survey of Protocol Reverse Engineering Algorithms: Decomposition of Tools for Static Traffic Analysis[J]. IEEE Communications Surveys & Tutorials, 2018, 21(1): 526-561.

[8] OUSIRIMANEECHAI N, SINTHUPINYO S. Extraction of Trend Keywords and Stop Words from Thai Facebook Pages Using Character n-Grams[J]. International Journal of Machine Learning and Computing, 2018, 8(6): 589-594.

[9] LIN M S, HAN X J, SONG W, et al. Based on multi-thread and multi-factor weighted keyword extraction algorithm[J]. Computer Engineering and Design, 2013, 34(7): 2398-2402.

[10] KRUEGER T K N P. Protocol Inspection and State Machine Analysis[J]. Journal of the American Chemical Society, 2014, 98(25): 8101-8107.

[11] ZHANG Z, ZHANG Z, LEE P P, et al. Proword: An unsupervised approach to protocol feature word extraction[C]// IEEE INFOCOM 2014-IEEE Conference on Computer Communications. 2014: 1393-1401.

[12] LUO J Z, YU S Z. Position-based automatic reverse engineering of network protocols[J]. Journal of Network and Computer Applications, 2013, 36(3): 1070-1077.

[13] HONG Z, TIAN Y F, ZHANG H Z, et al. Extended prefix tree based protocol format inference[J]. Computer Engineering and Applications, 2018, 54(12): 19-25.

[14] HOU F J, WANG L, WANG S, et al. Position-based automated protocol reverse engineer on network flows[J/OL]. Computer Engineering. https://doi.org/10.19678/j.jssn.1000-3428.0050950.

[15] ERMAN J, ARLITT M, MAHANTI A. Traffic classification using clustering algorithms[C]// Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data. 2006: 281-286.

[16] NIYAZMAND T, IZADI I. Pattern mining in alarm flood sequences using a modified PrefixSpan algorithm[J]. ISA Transactions, 2019, 90: 287-293.

[17] LI Y, LI Q, ZHANG X. Separate Protocol Message-Based Format Signature Construction Method for Variable Field[J]. Journal of Information Engineering University, 2018, 19(1): 30-38.

[18] PARK S H, SYNN J, KWON O H, et al. Apriori-based text mining method for the advancement of the transportation management plan in expressway work zones[J]. The Journal of Supercomputing, 2018, 74(3): 1283-1298.

CHEN Qing-chao, born in 1996, postgraduate. His main research interests include cyber security and so on.

WANG Tao, born in 1964, Ph.D, professor. His main research interests include cyber security and cryptography.