一种基于特征值的短信过滤匹配算法①

钱苏林，李 炜，王 晶

(北京邮电大学 网络与交换技术国家重点实验室，北京 100876)
(东信北邮信息技术有限公司，北京 100191)

摘 要：短信营销业务是近年来发展较快的短信业务之一，而随着该项业务的广泛应用，对短信的过滤匹配功能也提出了更高的要求。结合短信营销业务的特性，提出了一种高效的短信过滤匹配算法。算法依据短信分片特征值进行短信过滤，采用位向量法、编辑距离算法进行短信匹配，实现了高效的海量短信过滤匹配功能。文中给出了特征值算法的详细步骤，并对算法的有效性和正确性进行了分析对比。

关键词：短信；特征值；编辑距离；高效过滤匹配

Short Message Filtering and Matching Algorithm Based on Eigenvalues

QIAN Su-Lin, LI Wei, WANG Jing

(State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)
(EBUPT Information Technology Co.Ltd, Beijing 100191, China)

Abstract: SMS marketing is growing at a fast rate in the past few years. As this business is widely used, the requirement for the filtering and matching function of the short message is becoming increasingly higher. Having considering some characteristics of SMS advertising, we present an algorithm for effective text filtering and matching in this article. More specifically, the algorithm is briefly described as followed: firstly, text is filtered based on its eigenvalue and secondly using bit-vector method and Levenshtein Distance algorithm to match two candidate short messages thereby realizing the efficiency of mass short message filtering and matching function. This paper describes the details on calculation of text eigenvalue and presents the analysis on the effectiveness and correctness of this algorithm.

Key words: short message; eigenvalues; Levenshtein Distance; effective match and filter

1 引言

短信营销业务是近年来智能网发展较快的业务之一[1]，它为短信的业务量增长提供了一个很好的平台和解决方案。该业务向用户提供订购/点播短信的业务，当用户订购/点播之后成为种子用户，短信营销平台向用户下发种子短信，并由种子用户向其他用户转发，再由转发用户进一步向其他用户传播，从而形成多级转发链。由于种子用户可以从每一次转发行为中获益，所以就必须对每一条短信进行过滤匹配，以确认是否为种子用户的转发短信。在一个普通的短信营销业务应用案例中存在 50 万条种子短信、4 万/秒的短信流量，为了不存在积压，那么必须在一秒内完成 50 万*4 万=200 亿次短信内容的比较。所以，本文的研究重点是如何对海量的相似短信进行高效过滤和匹配，即如何提高字符串相似度的计算效率。

在文献[2-6]中提出了不同的计算字符串相似度算法，并且都尽可能的根据不同的应用场合提高计算效率，但这些算法都侧重于确定的两个字符串之间的匹配，而且都没有考虑到短信自身的特性，无法在海量的数据下实现高效的短信相似匹配。所以，本文结合短信自身的特性，针对如何提高海量短信的相似匹配效率问题进行了分析讨论，并提出了一种可行的高效的解决方案。

① 基金项目:国家自然科学基金(61072057,60902051);国家 973 计划(2012CB315802);中央高校基本科研业务费专项资金(BUPT2009RC0505);国家科技重大专项(2011ZX03002-001-01,2011ZX03002-002-01)
收稿时间:2011-08-15;收到修改稿时间:2011-09-14

2 术语定义

种子短信：短信跟踪平台需要进行跟踪的种子短信内容；

用户短信：从短信网关接收到的用户间发送的点到点短信；

种子匹配度：用户短信与种子短信相似程度的标杆，计算方式为用户短信和种子短信间的编辑距离[7]/用户短信长度和种子短信长度中的较大值*100%。

短信分片：每条短信根据标点符号进行分割，分割后的每个短信内容叫一个短信分片。每条短信都有若干条短信分片按照顺序组合；

MD5 密文：将短信分片进行 MD5 加密。MD5 加密后密文固定 32 个字节，并且具有不可抵赖性，也即修改加密前的种子内容的任何一位都会导致 MD5 加密密文不一致。MD5 加密密文不可解密（当然目前已经有解密算法了，但是极其复杂，并且效率低下）

特征值：每个短信分片的唯一的特征值。确定的短信有确定的特征值，但是确定的特征值对应很多个短信内容；

特征值种子：用于计算特征值的种子，每个短信特征值由 N 个特征值种子组成，可以设定 32 位机器 N 取 4，64 位机器 N 取 8。

3 短信格式分析

3.1 短信内容格式

根据短信网关上报的短信内容格式来看，种子短信匹配分为两种[8]：

MD5 分段加密密文：将短信分片进行 MD5 加密。

明文短信内容：短信内容采用明文方式传送。

3.2 短信内容格式分析

短信内容如果为分段（使用逗号分段）的 MD5 加密密文，那么在进行短信匹配时只能是全匹配，因为只要每段短信内容修改任何一个字符，均会改变密文，而且 MD5 加密密文为不可反转的，所以不能考虑将 MD5 加密密文转换成明文后再进行匹配分析。

而短信内容如果采用明文的方式进行传送，那么进行短信匹配分析的方法就可以更灵活，因为一般情况下对短信内容的修改都是在短信内容头或者尾加入或者删除一些内容，短信内容的主干部分并不会进行过多的修改。所以当短信内容使用明文进行传送时，同样可以考虑先对明文进行分片，然后对每个分片单独匹配，最后综合各个分片的匹配结果给出短信匹配结果。对单独的短信分片进行匹配时，考虑到用户可能对分片头或分片尾进行修改，所以应该去除分片头和分片尾来分析。

综上所述，分段短信的匹配均是采用一次或者多次全字符串匹配的方式，也即等于的方式，而不是查找。该类字符串匹配算法，已经被业界研究的很透彻了，所以我们应该转变思路，从其他的方向来提高短信过滤匹配的效率。

4 短信特征值的提出

4.1 处理能力瓶颈和短信特征值提出

字符串匹配的效率远远低于整形比较的效率，而字符串的匹配方法都已经比较成熟，因此为了提高处理能力，不能从种子短信和用户短信的字符串匹配效率入手，而是从匹配次数入手。也即需要从两个方面入手，首先是在字符串匹配前尽量的过滤需要匹配的用户短信个数，另外一个方面就是对于一个用户短信减少需要与其进行匹配的种子短信的个数。

在文献[9]中提出了一种基于区域特征的字符串相似过滤匹配算法，但该算法在选取区域特征时，只是随机的选取字符串的不同区域进行处理，而本文所研究的字符串主要就是短信。短信本身存在易分片，修改区域特定等特点。如果直接根据文献[9]中算法进行处理，那么在处理短信字符串的时候，就失去了进一步提升性能的可能性。

因此我们结合短信易分片，修改区域特定的特点，提出了利用短信特征值来实现快速过滤匹配短信的概念。主要方法是首先系统运行前计算出每个种子短信的特征值，运行期间计算出每条用户短信的特征值。根据该特征值首先检索出符合该特征值的种子短信集合。最后在该种子短信集合内进行字符串匹配。当然此时进行字符串匹配时，可以考虑短信内容做出了少量修改的匹配算法（MD5 加密方式除外）。

4.2 特征值提取方案

如图 1 所示，短信特征值集合的计算方法如下：

步骤 1. 将一条短信按标点符号分割成若干个短信分片，所述若干个短信分片按其在短信内容中的先后位置次序，依次排列成一个短信分片队列。

步骤 2. 对所述短信分片队列中的短信分片按长度从大到小的顺序进行排序，并从短信分片队列中顺序

提取 N 个最大长度的短信分片，所述 N 是特征值容量，可以根据主机型号来设置，32 位主机 N 可取 4，64 位主机 N 可取 8。例如 32 位主机下，短信分片内容为 A12345,B123456，C1234567，D12345678，E123456789，对短信分片按长度进行排序后，从最大长度的短信分片算起，所提取的 4 个短信分片依次为 E123456789、D12345678、C1234567、B123456。本算法对明文短信和 MD5 加密短信都可以提供支持，对于分片长度均为 32 个字节的 MD5 加密短信来说，本算法可以从短信分片队列中选取前面的 N 个短信分片，其匹配步骤和对明文短信的匹配步骤一致，以下就不再赘述。

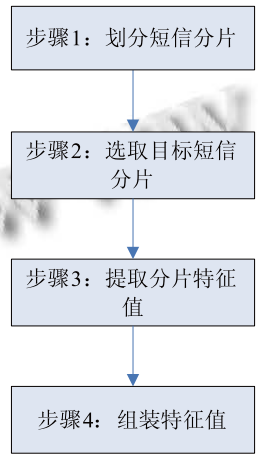

图 1 短信特征值计算流程

步骤 3. 按设定的字节提取位置，分别从所述 N 个短信分片的相应位置处提取字节作为特征值种子，对应于同一个字节提取位置的 N 个特征值种子构成一个特征值种子组，从不同的字节提取位置提取的特征值种子属于不同的特征值种子组。可以设定字节提取位置为每个短信分片的第一个字节；由于在短信发送过程中，用户很可能对短信内容进行编辑，比如增加抬头和敬语等相对比较短的语句，或者对个别的分段进行修改，因此也根据实际情况设定字节提取位置为每个短信分片的第一个字节和最后一个字节，同时提取短信分片的第一个字节和最后一个字节作为特征值种子，即第一个字节和最后一个字节所对应的特征值种子分别构成两个特征值种子组。

步骤 4. 将 N 个字节组合成一个整形数，该整形数即本条短信特征值。

举例对以上步骤进行说明：使用 32 主机，N=4，当收到短信：A12345,B123456，C1234567，D12345678，E123456789 时，首先对划分后的短信分片按长度进行排序，提取的 4 个短信分片依次为 E123456789、D12345678、C1234567、B123456；再同时提取短信分片的第一个字节和最后一个字节为特征值种子，即两个特征值种子组分别为 E、D、C、B 和 9、8、7、6；其 ASCII 码分别为 0x45，0x44，0x43，0x42 以及 0x39，0x38，0x37，0x36。则该短信的特征值为 0x45444342 和 0x39383736。

当短信不足 N 个分片时，可以循环取下一列的字节进行补足 N 个字节。如短信内容为 A54321,B654321,C7654321 时，短信分片不足 4 个，按照分片长度进行逆排序，先取出 C、B、A，再取分片 C7654321 的下一个字节进行补足，如此条短信取 7 进行补足。所以该短信的特征值为 0x43424137。若取完 7 后仍不足 N 个字节，则继续取 B654321 分片的下一个字节进行补足，此处取 6。

4.3 改进的特征值集合

一般在转发短信时，很可能加上“XX 您好：”或者“。XX 上”的内容。在进行短信明文匹配时加上该短信不会影响其特征值（只取最长的短信内容），但是在使用 MD5 分片的时候，各分片长度一致。而且可能存在用户修改了某个分片，而没有加入标点符号。因此提出特征值集合的概念。

根据 4.2 的特征值提取方案，每个短信可以生成一个或多个含有 N 字节的特征值种子组，从每个特征值种子组中任选 M 个特征值种子，并对所述 M 个特征值种子按正序进行排列，将排列后的特征值种子对应的 ASCII 码组合为一个整形数，这些整形数就是特征值集合，而每个整形数即新的特征值。其中 M 是特征值集合容量，也可以根据主机型号来设置，例如 32 位主机 M 取 3，64 位主机 M 取 4。

继续 4.2 的特征值提取步骤，从特征值种子组{EDCB}和{9876}中任选 3 个特征值种子，并按正序排列组成一个特征值，所述特征值集合记为{EDC，EDB，ECB，DCB，987，986，976，876}。对于第一个特征值 EDC 来说，其逆序排列为 E、D、C，其对应的 ASCII 码分别为 0x45、0x44、0x43，则在 32 位主机上该短信的特征值为 0x454443，对于不足 4 个字节的左补零。

当使用 64 主机时，还可以去掉更多个特征值种子，如可以最多去掉 4 个种子，也即从 8 个特征值种

子中任选 4 个特征值种子组合成为特征值集合。因此一条短信对应于 70*2 个特征值的特征值集合。

引进特征值集合的概念后，32 位主机下，一条短信特征值个数会从 2 个变为 8 个，虽然会增加 4 倍的匹配和查询次数，但是显著提高了特征值的准确率，降低了丢失率。

4.4 短信特征值的特点

1. 唯一性：通过特征值的选取可以看出对于确定的短信，其特征值集合唯一确定。

2. 易算性：特征值选取无需进行复杂操作，按照指定位数进行选取，并且进行位操作即可，而位操作效率非常高。

3. 高效性：特征值集合中的每个特征值均为一个整形数，计算机在进行操作中，对 8 位和 64 位的类型进行比较和操作的效率一致，也即对 64 位的长整型的比较和对一个字符的比较效率一致，从而远远高于字符串的比较效率。

4. 离散性：64 位主机下，特征值可以达到 8 位，按照从 8 个特征值种子中取 4 个特征值种子构成特征值集合的方式，其容量为 2 的 32 次方也即达到 42 亿，即使考虑到每条个短信的特征值集合有 70 个特征值，那么也有 6 千万的种子短信的容量。

5. 非线性：理论数据上特征值的值域达到 256 的 4 次方，即 40 亿，每条短信有 70*2=140 个特征值，也即平均可以容纳 2.8 千万的种子短信，也即是说，理论上在 2.8 千万以下的种子短信库的情况下匹配次数为 1，或者说和种子短信库容量无关。

5 短信特征值过滤匹配算法

5.1 种子短信库初始化流程

如图 2 所示，种子短信库初始化流程分以下步骤进行：

步骤 1. 计算种子短信库中每条种子短信的特征值集合，并将所述种子短信的短信内容和特征值保存。可将种子短信的特征值和短信内容保存在 B+树的排序容器中，由于每条短信最多对应于 70*2 个特征值，可以使用一个种子短信节点队列保存种子短信的详细信息，且每个种子短信对应于种子短信节点队列中的一个节点，特征值 B+树保存特征值和其对应的种子短信的指针地址；

步骤 2. 计算种子短信的最小匹配长度：即将种子短信长度乘以种子匹配度。所述种子短信的最小匹配长度，是在达到种子匹配度要求的情况下，用户短信的最小长度。有些种子短信要求全匹配，即种子匹配度为 100%，那么种子短信的最小匹配长度就是种子短信长度；有些种子短信要求达到一定匹配度即可，那么种子短信的最小匹配长度就是种子短信长度乘以种子匹配度；

步骤 3. 比较所有种子短信的最小匹配长度，挑选出其中的最小值，将所述最小的种子短信的最小匹配长度记为 Lmin。

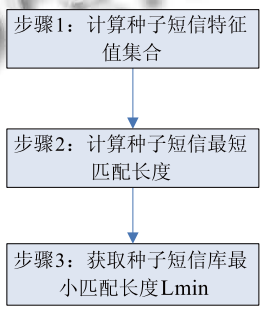

图 2　种子短信初始化流程图

5.2 用户短信处理流程

如图 3 所示，用户短信处理流程分以下步骤进行：

步骤 1. 第一次长度过滤：判断所述用户短信长度是否小于 Lmin？如果是，则没有和用户短信相匹配的种子短信，本流程结束；如果否，则继续下一步骤 2。

步骤 2. 计算用户短信的特征值集合。所述方法流程参见 4.2 小节。

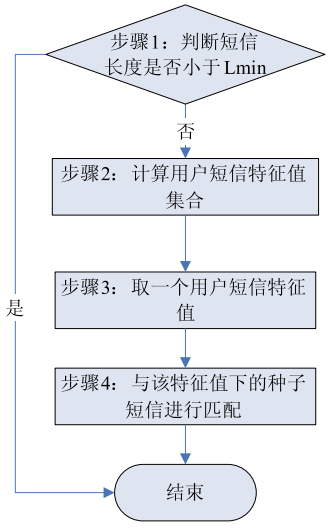

图 3　用户短信处理流程图

步骤 3. 从特征值集合中取一个特征值；

步骤 4.将用户短信与该特征值下的种子短信进行匹配。

5.3 用户短信与候选种子短信匹配流程

如图 4 所示，用户短信与候选种子短信匹配流程分以下步骤进行：

步骤 1. 从用户短信的特征值集合中提取一个特征值。

步骤 2. 判断种子短信库中是否有和所述特征值一致的种子短信？如果是，则继续以下步骤；否则，结束该流程。

步骤 3. 第二次长度过滤：判断用户短信的长度是否不小于种子短信的最小匹配长度，且不大于种子短信的最大匹配长度，其中，种子短信的最小匹配长度=种子短信长度×种子匹配度，种子短信的最大匹配长度=种子短信长度+种子短信长度×（1-种子匹配度）？如果是，则继续下一步骤；否则，结束该流程。

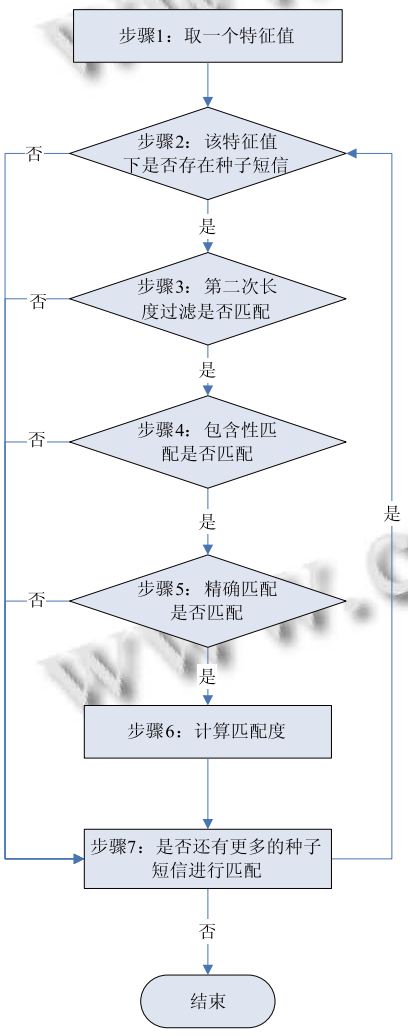

图 4 用户短信与候选种子短信匹配流程图

步骤 4. 包含性匹配：首先不考虑用户短信和种子短信相同内容的匹配顺序问题，在用户短信中查找种子短信每个字符，然后计算匹配度，如果匹配度达不到最小要求进行过滤，不再进行后续计算。包含性匹配使用改进的位向量法进行运算[10]。

步骤 5. 精确匹配：使用改进的编辑距离法计算种子短信和用户短信的编辑距离[11]：编辑距离(Levenshtein Distance) 由 Levenshtein 于 1966 年在文献[7]中提出，通过编辑距离计算源字符串 S 与目标字符串 T 相似度。编辑距离是指由 S 变化到 T 所需要的最小编辑操作的数量，Levenshtein 所提出的编辑操作是指对字符串的某一个位置的字符进行删除、插入、替换的操作。文献[12]对编辑操作进行了扩展，增加了两相邻位置的字符间的交换操作，以实现编辑操作的最小化。借鉴文献[11]中的方法，通过计算得出将字符串 A 经过 N 步的操作即可变为字符串 B，来计算 A 和 B 的编辑距离，那么 A 和 B 的匹配度则为 N/max(length(A), length(B))。

步骤 6. 作为符合要求的匹配结果，将所述种子短信和根据编辑距离计算出的精确匹配度保存。

步骤 7. 判断种子短信库中是否还有和所述特征值一致的种子短信？如果有，则转向步骤 3；否则，结束本流程。

6 特征值过滤匹配算法分析

6.1 种子短信库分析

本文采用 365984 条字符串，在 64 位机器环境下，对种子短信库进行模拟，分别对特征值种子集合 8 选 4 和特征值种子集合 8 选 5 两种情况进行测试。表 1 为在不同特征值选取方案下，种子短信库的相关情况对比。

表 1 中的数据显示，在特征值集合容量 8 选 4 的方案下，特征值总个数少于特征值集合容量 8 选 5 的方案，而所有特征值下的种子短信节点个数之和反而多于特征值集合容量 8 选 5 的情况。这主要是由于在 8 选 4 的情况下，只从 8 个特征值种子中选 4 个，当所有种子短信特征值都计算出来后，存在很多相同的特征值，重复性较大；但是 8 选 5 的情况下，特征值种子的组合更多，相对的重复性就较小，所以特征值

总个数 8 选 5 的方案更多些。而由于 8 选 4 有 70 种的组合，8 选 5 有 56 种的组合，也就是说 8 选 4 的情况，一个种子短信节点可能会分布在 70 个特征值下，8 选 5 的情况，一个种子短信节点可能会分布在 56 个特征值下，之前的分析只是说明会过滤掉重复的特征值，而一个特征值所对应的种子短信节点并没有被过滤，所以 8 选 4 的方案下，所有特征值下的种子短信节点个数之和会更多。

表 1　不同方案下种子短信总体分布

	特征值 8 选 4	特征值 8 选 5
匹配度	80%	80%
特征值总个数	10498234	15119103
所有特征值下种子短信节点个数之和	21371595	17371029

但是从总体上来看，种子短信节点的分布还是很稀疏的，在特征值集合容量 8 选 4 的情况下，平均每个特征值下面只会对应 2.035 个种子短信，也就是说在最差的情况下，一条用户短信对应的 70 个特征值不重复，这 70 个特征值下面对应的种子短信也没有交叉，那么平均也只需要和 140 条种子短信进行匹配就可以计算出相似度，比整个库的大小：36 万条种子短信，已经提升了 1000 倍以上。在 8 选 5 的方案下，平均每个特征值下只有 1.15 个种子短信节点，匹配的次数会更少。

下面再看一下种子短信节点的细化分布。

表 2　不同方案下种子短信细化分布

	特征值 8 选 4	特征值 8 选 5
单个特征值下对应种子短信数最大值	1625	572
特征值个数（单个特征值下对应种子短信数=1）	7379069	13800130
特征值个数（单个特征值下对应种子短信数=2）	1497421	917167
特征值个数（2<单个特征值下对应种子短信数<=50）	1610424	401678
特征值个数（50<单个特征值下对应种子短信数<=100）	9213	121
特征值个数（100<单个特征值下对应种子短信数<=200）	1868	6
特征值个数（200<单个特征值下对应种子短信数<=max）	239	1
特征值总个数	10498234	15119103

如表 2 所示，虽然有个别特征值下短信会非常聚集，如 8 选 4 的方案下，单个特征值下对应种子短信数最大值达到了 1625 个种子短信节点，但绝大多数的特征值下只会存在很少量的种子短信节点，在 8 选 4 的方案下，84%以上的特征值对应的种子短信节点少于 2 个，而且其中大多数特征值只对应一个种子短信节点。特征值对应的种子短信节点个数超过 100 的特征值仅占很小的一部分。8 选 5 的方案则更优，不光聚集的种子短信节点最大值更小，只有 572 个，而且 97%以上的特征值对应的种子短信节点少于 2 个，绝大多数只对应一个节点。

经过以上分析可以说明特征值算法对整个种子短信是具有很强的离散作用的，从理论上就已经获得了很好的结论，而在实际测试过程中，结果比预计的还要好。

6.2 特征值算法有效性分析

选取随机的 26565 短信进行测试，并插入一定数量的相似种子短信以能进行匹配。种子短信库选择 6.1 所模拟的种子短信库，机器位数 64 位。如表 3 所示为在 64 位机器下，同种子匹配度下特征值种子 8 选 4 与 8 选 5 情况的算法测试结果。

表 3　同匹配度下有效性实验结果

	特征值 8 选 4	特征值 8 选 5
匹配度	80%	80%
用户短信个数	26565	26565
最短用户短信长度	41	41
第一次长度过滤掉的用户短信个数	13231	13231
剩余用户短信个数	13334	13334
第二次长度过滤掉的种子短信个数	2726177	247976
特征值集合个数	660478	543408
平均特征值集合个数	49	40
总包含性匹配次数	957504	82534
平均包含性匹配次数	71	40
总精确匹配次数	10594	8636
平均精确匹配次数	0.79	0.65
匹配的用户短信个数	52	52

从实验结果可以看出，若不使用特征值算法，那么除了第一次长度过滤掉的 13231 条用户短信，余下的 13334 条用户短信都要与种子库中的每一条种子短信进行存在性匹配和精确匹配，也就是说每一条用户短信需要进行 36 万次匹配。而使用了特征值算法，在 8 选 4 的情况下，平均每一条用户短信只需要进行 71

次存在性匹配和 0.79 次精确匹配，效率提高了 1 万倍。

再对比不同匹配度下特征值算法的表现，如表 4 所示为同样的特征值种子选取方案下，不同的种子匹配度情况的算法测试结果。

表 4 同选取方案下有效性实验结果

	特征值 8 选 4	特征值 8 选 4
匹配度	80%	90%
用户短信个数	26565	26565
最短用户短信长度	41	46
第一次长度过滤掉的用户短信个数	13231	13838
剩余用户短信个数	13334	12727
第二次长度过滤掉的种子短信个数	2726177	3059875
特征值集合个数	660478	627320
平均特征值集合个数	49	49
总包含性匹配次数	957504	452542
平均包含性匹配次数	71	35
总精确匹配次数	10594	8313
平均精确匹配次数	0.79	0.65
匹配的用户短信个数	52	19

同为 8 选 4 的方案，在将匹配度从 80%提高到了 90%之后，性能也得到了较大提升，不仅每次长度过滤都能过滤掉更多的短信，而且平均包含性匹配次数仅为原来的一半，平均精确匹配次数也从原来的 0.79 次降为了 0.65 次。当然提高了匹配度之后，匹配上的用户短信个数会有一定的降低，实验中，匹配的用户短信数从 52 条降到了 19 条。

同时，从实验结果中可以分析得出，种子短信库的大小会影响系统性能，库越大，需要匹配的种子短信越多，性能会降低。特征值集合个数的选取会很大程度影响性能，如表 3 中显示，在 8 选 4 的方案下，平均进行的包含性匹配次数为 71 次，平均进行的精确匹配次数为 0.79 次，而 8 选 5 的方案下平均包含性匹配次数为 40 次，平均进行的精确匹配次数为 0.65 次，在同时匹配上 52 条用户短信时，8 选 5 的方案性能有所下降。

但是结合 6.1 节的分析，若当匹配度要求不高时，一味的选择更多的特征值种子进行特征值的拼装，可能会使得算法结果存在偏差。对如下情况进行分析，假设一条用户短信经过多次转发后，依然符合算法的匹配度要求，假设用户短信的特征值种子组为：ABCDEFGH，与它匹配的种子短信特征值种子组为：ABCD1234，若在这种情况下，算法依旧选择 8 选 5 的方案，那么就会导致该用户短信无法正确进行匹配，因为它们的特征值种子集合没有交叉，也就是说算法不会将该用户短信与该种子短信进行相似度计算，造成偏差。

当然，这种情况仅仅是理论上存在的一种可能，实际情况中发生的概率非常小，因为如果特征值种子组相差这么大，那么改动的短信内容也会较多，就可以认为这两条短信不相似，然后通过提高匹配度使之成为不匹配的用户短信。

6.3 特征值算法准确性分析

实际的短信营销业务主要包括有以下几种情况：

（1）对于没有改动过的用户短信来说，种子短信和用户短信完全一致，其特征值也完全一致，因此其正确度为 100%；

（2）对于改动过的用户短信来说，种子短信一般为多分片的祝福短信，如“劳累了一年的您该歇歇了，今天是父亲节，我作为您的孩子，已经成长为一个大人了，不会再让您为我如此操心了，祝愿您节日快乐，永远快乐”，“我知道您对我的爱像太阳，只是您以月亮的方式向我表达；我知道您对我的爱像大海，只是您以小溪的方式向我表达；今天是父亲节，愿爸爸幸福安康。”等等，均为 4 个分片以上的，尤其是排比句式的祝福短信。用户的实际短信可以细分为以下几种情况：

a. 在实际发送过程中，绝大部分改动均是加上称呼头，或者姓名落款，如“XX 您好：”。这只是增加了分片个数，没有改动原来的分片的内容，因此也不会改变其对应的特征值，也即正确率为 100%；

b. 如果在短信内容中修改了短信内容，由于特征值种子从短信分片头中提取，因此不会影响特征值计算，也即正确率为 100%；

c. 对于没有加标点，而且修改了超过 5 个（含）分片的短信，并且短信头尾均有改动的情况下，才会出现低于 100%的准确率。但是此时已经至少修改了 10 个字符，而一条短信也就 70 个汉字。也就是说其最高的匹配度也只有 85%了，可以不识别为种子短信。

7 总结

本文提出了一种高效的短信过滤匹配算法，它通

过对短信分片，选取计算特征值，进行快速的短信定位过滤，并采用位向量法和编辑距离算法计算短信间的相似度，同时确保短信过滤匹配算法的高效性和正确性。

该算法需要对种子短信库进行初始化，保存种子短信的特征值及索引，使用了额外的内存空间，但通过特征值过滤机制，有效的减少了短信匹配次数，提高了海量短信过滤匹配的速度。因此，从实际应用的角度出发，本算法为短信营销业务的推广和扩大提供了一种高效的、可行的短信过滤匹配方案。

参考文献

1 廖建新.移动智能网技术的研发现状及未来发展.电子学报,2003,31(11):1725−1731.

2 车万翔,等.基于改进编辑距离的中文相似句子检索.高技术通讯,2004,14(7):15−19.

3 邹旭楷.汉字/字符串编辑距离和编辑路径的有效求解技术.计算机研究与发展,1996,33(8):574−580.

4 王斌,等.支持块编辑距离的索引结构.计算机研究与发展,2010,47(1):191−199.

5 薛晔伟,等.一种编辑距离算法及其在网页搜索中的应用.西安交通大学学报,2008,42(12):1450−1454.

6 李彬.计算字符串相似度的矩阵算法.现代电子技术,2007,30(24):106−111.

7 Levenshtein VL. Binary codes capable of correcting deletions, insertions and reversals. Doklady Akademii Nauk SSSR, 1966,163(4):707−710.

8 Zhang YT, Liao JX, Zhang TY, Zhu XM. A novel method for the short message or multimedia message synchronization. Second International Conference on Wireless and Mobile Communications. 2006.10.

9 孙德才等.基于匹配区域特征的相似字符串匹配过滤算法.计算机研究与发展,2010,47(4):663−670.

10 陈开渠,赵洁,彭志威.快速中文字符串模糊匹配算法.中文信息学报,2004,18(2):58−65.

11 赵作鹏.一种改进的编辑距离算法及其在数据处理中的应用.计算机应用,2009,29(2):424−426.

12 Levenshtein VL. Binary codes capable of correcting deletions, insertions and reversals. Doklady Akademii Nauk SSSR, 1966,163(4):707−710.

13 Lowrance R, Wagner RA. An extension of the string to string correction problem. Journal of the ACM, 1975,22 (2):177−183.

(上接第 41 页)

3 潘巨龙,李善平,吴震东.基于无线传感器网络的社区保健监测系统.中国计量学院学报,2007,18(2):136−150.

4 唐明霞,王秋光.独居老人无线监护系统用户端的设计.哈尔滨理工大学学报,2006,11(6):49−52.

5 郑凯,赵宏伟,张孝临.基于 Zigbee 心电监护网络的定位系统的研究.仪器仪表学报,2008,29(5):1000−1005.

6 刘自立,许剑,陈金水,等.基于无线片上系统技术的心电监护系统的设计与开发.中国科技论文在线,2009,4(1):69−74.

7 潘健勇,董齐芬,潘浩,俞立.基于无线传感网的远程心电监护系统研究与设计.东南大学学报(自然科学版),2010,(S1):161−171.

8 Lee S, Lee M. A real-time ECG data compression algorithm for a digital holter system. Proc. Engineering in Medicine and Biology Society. Vancouver, Canada, 2008:4736−4739.

9 Li NQ, Li P. A Range-Free Localization Scheme in Wireless Sensor Networks: Knowledge Acquisition and Modeling Workshop. 2008:525−528.

10 Simic SN, Sastry S. Distributed localization in wireless Ad Hoc networks, UCB/ERL/M02/26, 2002.

11 He T, Huang C, Blum BM, et al. Range free localization schemes for large scale sensor networks. Proc. of the 9th Annual International Conference on Mobile Computing and Networking (Mobi Com). San Diego, CA, 2003:75−100.

12 Patwari N, Ash JN, Kyperountas S, et al. Locating the nodes cooperative localization in wireless sensor networks. IEEE Signal Processing Magazine, 2005,22(4):54−69.