利用 Java 语言处理俄语问题初探

许汉成

（国防科技大学国际关系学院，南京 210039）

提　要：本文通过实验方法研究了 Java 语言处理俄语的能力，重点展示了 Java 的 String 类及 Java 正则表达式在俄语信息处理方面的应用。作者认为，考虑到中文平台输入和输出中俄文混文本的实际需要和困难，Java 值得有志于从事俄语计算研究的学者投入时间精力。

关键词：俄语；自然语言处理；Java；String 类；正则表达式

中图分类号：**H35**　　**文献标识码：A**

1 引言

自从进入信息社会，人类就面临着海量信息处理的问题，而文本始终是信息的主要载体。文本大多是用自然语言书写的，这当然包括俄语文本。利用计算机自动化处理俄语、俄汉双语问题的解决速度、时间、质量，体现了我们这个国家在互联网时代处理海量俄文信息的能力，涉及国家安全和社会、经济发展。

俄语、俄汉双语的自动化处理，与英语和英汉双语的自动化面临的形势不同：国内程序员多数多多少少懂些英语，处理起英语、英汉双语问题比较得心应手。英语又是国内最流行的外语，市场庞大，公司、研究者愿意投入时间、精力和各种资源，而国际上英语自动化处理起步早、开放性好、从业者众，语料库、算法、程序随处可见。俄语、俄汉双语的处理就困难得多，面临着完全不同的环境，连正常的教学和科研平台都没有。因此，俄语单语、俄汉双语的自动化处理问题还是要依靠国内俄语学者自身的努力。一些与俄语语言特点相关的计算问题，只能靠俄语学者转换思维方式，学习掌握一定软件工程知识，才能彻底解决，至于俄汉双语的问题，连俄罗斯人也都不能指望，他们的研究重点不会是汉语。

二十世纪末起，计算语言学已经突破了形式语法的框框，更多地研究大规模文本的自动检索、抽取、分类、聚类等问题，解决这些问题的必要条件是具备语料库、统计模型、算法，熟练使用一门程序语言是重中之中，是打开俄汉语自动化处理的金钥匙。俄语的编码复杂，在汉语平台上输入、输出俄汉混合文本面临许多挑战。个人在 Windows 和 Linux 系统上尝试过多种编程语言，从 Lisp、Prolog、Basic/VB、C/C++、Perl、Python 到 Java、C#。经过多年摸索，个人认为，就俄汉语处理的程序语言工具而言，做 Windows 桌面程序，C#是最好的选择。Python 入门简单，文科生容易掌握，功能也比较强大，是很好的教学语言。从开发资源、潜能和综合性能出发，Java 是很好的选择。Java 是一种成熟的、面向对象的编程语言，该语言在设计之初便考虑了多语言处理问题，系统内部采用 Unicode 表示字符，可以方便地处理世界上各种语言，而且具备跨平台运行能力。本文主要探讨 Java 的俄语处理能力。

2 实验环境

语言信息处理有高深的理论，如机器学习、统计语言模型，是目前软件科学研究的热点。但是，俄汉处理必须脚踏实地，从基础开始，多上机实践，才能打好俄汉语处理基础。俄汉语处理的实验环境包括很多东西，基础是计算机硬件、操作系统和程序语言。为了实现俄汉处理的目标，我们这里采用的操作系统是 Window 7 中文版、JDK 1.7.0，编程工具就使用 JCreator 4.50（也可以使用纯文本编辑器 Notepad++）。

3 俄语字符串的处理

计算机识别自然语言与人理解语言文字完全不同。计算机内部用二进制数字表示和存储字符（包括字母、数字、标点符号、控制符等），而人所看到的词、句子、文本在计算机里不过是二进制数字串，或者说是字符数组。Java 语言库函数提供了大量与文本处理相关的类，如字符（Character）、字符串（String）、字符集（Charset），其中 String 类可以直接用于俄语字符的存储和处理，其重要性不言而喻。这里最主要研究这个类。另外，我们还需要用 javax.swing.JOptionPane 类的 showInputDialog 函数来获取俄语单词、句子，用 javax.swing.JOptionPane 的 showMessageDialog 来显示处理结果。

俄语字符的编码方案很多，无论是哪个编码，Window 系统编（cp1251）码的俄语字母的值在 128 – 255 之间，排在拉丁字母、西文标点符号、控制符之后，要用到 8 位、1 个字节的最高位。汉语的编码也要用 8 位的最高位。俄汉处理中的乱码现象多由此产生。Java 内部采用 Unicode 表示字符，计算一个字符串的长度时，会返回字母个数，而不是字节数，与语言学家的直觉相同。下面程序利用 Java 的 String 类的函数 length()来计算字符串的长度：

```
import javax.swing.JOptionPane;

public class cyrStrLen {

public static void main(String[] args){

        String str = JOptionPane.showInputDialog("请输入字符串：");

        int    len = str.length();

        JOptionPane.showMessageDialog(null, str + "的长度为: " + str.length());

        System.out.println(str);

        System.out.println(str + "的长度为: " + str.length());

  }

}
```

我们输入 хороший，得到“хороший 的长度为：7”。 Java 在计算汉语词语的长度时，也会以汉字为单位的。我们还可以看到，在对话框里，汉字与俄语的显示非常正常。Java 的这些特性，减轻了学习俄汉语处理问题的入门难度。

Java 字符串类的 charAt 函数让我们可以方便地通过下标取到俄语单词的每一个字母，我们可以进一步判断，这个俄语字母是大写还是小写，是元音字母还是辅音字母。下面两行代码可以输出俄语单词的首、尾字母：

```
System.out.println(str + "的第一个字符为" + str.charAt(0)); //第一字符

System.out.println(str + "的最后一个字符为" + str.charAt(str.length()-1)); //最后一个字符
```

其中，str 是我们用来存储输入字符串的变量。何淑琴（2005）给出过几个测量文本难易程

序的公式，我们完全可以借鉴英语的经验，计算俄语文本的难度。要完成这个任务，就需要计算词长、句长、音节数等。下面代码通过计算元音字母个数计算俄语单词元音字母和辅音字母个数，可以成为解决上述问题的基础：

```
int num_vow=0, num_con=0;

    String cyrStr = JOptionPane.showInputDialog("请输入一个俄语字符串：");

    int len = cyrStr.length();

    for(int i=0; i<len;i++)

    {

        char c = Character.toLowerCase(cyrStr.charAt(i));

        //俄语元音字母

        if(c=='а'|| c=='и'||c=='о'||c=='у'||c=='э'||c=='ы'

           ||c=='е'||c=='ё'|| c=='ю'||c=='я')

        {

            num_vow++;

        }

        //俄语辅音字母

        if(c=='б'|| c=='в'||c=='г'||c=='д'||c=='ж'|| c=='з'||

            c=='й'||c=='к'||c=='л'||c=='м'||c=='н'|| c=='п'||

            c=='р'||c=='с'||c=='т'||c=='ф'||c=='х'|| c=='ц'||

            c=='ч'||c=='ш'||c=='щ'||c=='ь'||c=='ъ')

            {

              num_con++;

            }

    }

    JOptionPane.showMessageDialog(null,cyrStr + " 有 "+num_con+" 辅 音 字 母 ，"+num_vow+"个元音字母。");
```

俄语单词的结构包含前缀、后缀、词根、后缀等，我们可以用 Java 的 String 类 startsWith 和 endsWith 来判断一个生词的开头和结尾。下面我们先分析俄语词是否以 при 开头、以 ся 结尾，然后用 substring 函数去除 приходиться 的前缀和尾缀。

```
    String str = JOptionPane.showInputDialog("请输入字符串：");

    String temp = null;

    System.out.println(str);

    System.out.println(str + "是否以 при 开始：" + str.startsWith("при") );
```

```
        System.out.println(str + "是否以 ся 开始：" + str.endsWith("ся") );
        if(str.startsWith("при")){
            temp = str.substring("при".length());
            if(str.endsWith("ся")){
                temp = temp.substring(0,temp.length()-2);
               }
            System.out.println(temp);
            }
    }
```

toUpperCase 和 toLowerCase 可以用来转换俄语字母的大小写，例如把 путин 和 МОСКВА 变为 ПУТИН 和 москва。函数 CompareTo 可以用来比较字符串对象的大小（区分大小写），这个函数返回一个整数，两个字符串相等，返回 0，如果字符串对象大于函数参数表示的另一个字符串，则返回值大于 0，相反则小于 0。CompareToIgnoreCase 用于进行不区分大小写的比较。

```
System.out.println("区分大小写，Москва 与 москва 不相同：" + "Москва".compareTo("москва"));  //输出-32
System.out.println("不区分大小写，Москва 与 москва 相同：" + "Москва".compareToIgnoreCase("москва"));  //输出 0
System.out.println("приходиться 包含 ходить: " + "приходиться".contains("ходить")); //true
```

Выйти 的过去时阴性和复数形式，是由 выш 加 ла 和 ли 构成的：

```
System.out.println("выш".concat("ла"));  //输出 вышла
System.out.println("выш".concat("ли"));  //输出 вышли
```

indexOf 和 lastIndexOf 可以用来从开头和结尾查找特定字符串是否出现以及出现位置：

```
System.out.println("приходить".indexOf("ходить"));  //输出 3
System.out.println("вовремя".lastIndexOf("мя"));  //输出 5
```

替换操作在文本信息处理中具有重要意义，Java 的 String 类的 replace, replaceAll, replaceFirst 可以完成替换、全部替换、替换第一次出现目标的操作：

```
System.out.println("выйти".replace("вы","во"));  //输出 войти
System.out.println("пробовать".replaceAll("о","и")); //输入 прибивать
System.out.println("пробовать".replaceFirst("о","и")); //输出 прибовать[1]
```

对于字母文字，将由词形、标点符号、空格构成的连续文本以空白等分隔符号为词的标记切分成一个个词形的过程叫做分词（tokenization, токенизация）。Java 的 String 类的函数 split 可以实现分词任务。下面程序将 У Лукоморья дуб зеленый 切分成词，并计算每个词的长度：

```
public class ruToken {
    public static void main(String[] args) {
        String line = "У Лукоморья дуб зеленый";
```

```
            String words[] = line.split("\\s"); //以空白为标记分词
            for(int i=0;i<words.length;i++){
//trim 函数去除字符串前后的空白
                System.out.println(words[i]+ " : " + words[i].trim().length());
            }
        }
}
```

4 正则表达式与俄语文本处理

从计算机专业角度看，正则表达式（regualar expression）是一种描述具有共同特征的字符串集合的方法。从语言研究和教学的角度看，正则表达式提供了一种在一个或多个文本查找、编辑、替换多个字符串的强大、方便的手段。我们采用任务驱动的方式来解释这个问题。现在假定我们一份长达数千人的俄罗斯名人表，按照名、父称、姓的方式排列，下面是这张表的部分人名：

…

Александр Сергеевич Пушкин

Антон Павлович Чехов

Георгий Константинович Жуков

Михаил Илларионович Голенищев-Кутузов

…

现在有一个任务：要求将姓排在最前边，然后是名、父称，而且我们希望在姓后面插入一个逗号。如果手工一条一条纪录操作，不断地剪切、拷贝，恐怕需要花费很多时间。这个任务可以用正则表达式来完成，主要包括下面两步：

1） 用于查找、匹配人名的表达式：([А-Яа-я-]+)\s+([А-Яа-я-]+)\s+([А-Яа-я-]+)

2） 替换匹配对象的表达式：$3,$1 $2

然后执行相关命令，我们就可以得到下面结果：

…

Пушкин, Александр Сергеевич

Чехов, Антон Павлович

Жуков, Георгий Константинович

Голенищев-Кутузов Михаил Илларионович

…

如果会 Java 语言，我们可以写程序完成这个任务，不会也没有关系，有一个免费 Java 程序 JRegExAnalyser，可以测试[2]、分析正则表达式，很多编辑器（如 EmEditor、Notepad++、UltraEdit 都支持正则表达式。前面 Java 搜索和替换表达式体现一些正则表达式的部分要素：输入字符串、匹配和替换模式。我们利用匹配模式去输入字符串里查找目标字符串，如果能够匹配成功，我们可以进一步按照替换模式去替换匹配的输入字符串相关部分。java.util.regex 包描述支持的正则表达式主要语法规则如下：

正则表达式	含义
.	任意一个字符
[abc]	匹配由 a,b,c 组成的字符集之中任意字符

[^abc]	匹配除 a,b,c 以外的任意字符
[a-z]	A 到 z 的所有字符之一
\d	任意一位数字，即[0-9]
\D	数学以外一个字符，即[^0-9]
\s	空白字符，包括[\t\n\x0B\f\r]
\S	非空白字符，即[^\s]
\w	一个字符，即[a-zA-Z_0-9]
\W	非字符
X?	X，出现 0 次或 1 次
X*	字符 X 出现 0 或多次
X{n}	字符 X 出现 n 次
X{n,}	字符 X 至少出现 n 次
X{n,m}	字符 X 出现 n 至 m 次
^	在多行模式下匹配一行开头
$	在多行模式下匹配一行末尾
XY	X 后接着 Y
X|Y	X 或 Y
（X）	匹配 X 并其标注为一个组，以后可以引用
\n	第 n 组

正则表达式可以用来查找和处理俄语文本。例如，正则表达式“(ся|сь)$”可以找带 ся 动词，“\b(в|на)\b”可以找到 в 和 на 出现的各种场合，“[вВ]оенн.*{1,3}”可以找到 военный 的各种形式，包括 военно-учетные специальности 之中的 военно，есть|был[оаи]*|буд.{1,3}|быть 可以找到 быть 的各种形式。如果把概念看成表达概念的词的集合，那么只要正则表达式列举词及其各种词法形式，我们就能够在概念层面上对文本进行统计分析。

自然语言处理问题涉及大量模式识别问题，其中包括自动分词和断句，俄语词形态变化丰富，因此分词操作得到实际是词形（словоформа），而不是词的原形（лемма），句子是语言信息处理的另一个重要单位，正则表达式是解决问题的重要手段。下面举例说明一下俄语分词和断句的问题：

Мы хотим создать мощный военный научно-учебный центр, где будет совмещен и процесс обучения и процесс науки и адаптировано более практически к реалиям сегодняшнего дня. Мы предусматриваем, что часть должностей офицеров будут заменяться гражданским персоналом, в том числе военные юристы, финансовые работники будут переходить в большей степени на гражданские специальности. Мы за счет громадного количества освобождаемых должностей офицеров можем в разы увеличить заработную плату и денежное содержание гражданского персонала и офицеров.

我们搜索正则表达式“\.”，将它替换为“\.\n”，即在所有句号之后一个换行符，这样可以得到下面几个句子：

• Мы хотим создать мощный военный научно-учебный центр, где будет совмещен и процесс обучения и процесс науки и адаптировано более практически к реалиям сегодняшнего дня

• Мы предусматриваем, что часть должностей офицеров будут заменяться гражданским персоналом, в том числе военные юристы, финансовые работники будут переходить в большей степени на гражданские специальности

• Мы за счет громадного количества освобождаемых должностей офицеров можем в разы увеличить заработную плату и денежное содержание гражданского персонала и офицеров

如果我们用查找正则表达式“\s”，即空白（空格，换行符，制表符等），并且替换成“\n”，那就可以将上面短文切分成 138 个词形，除了个别标点符号紧挨着词形的问题外（如“центр,”），似乎效果还可以。其实只要对上面短文先进行预处理，比如先利用正则表达式，搜索所有标点符号并在其前面添加一个空格（搜索(“([\.,!?;])”，替换为“ \1”），然后再将空格替换为换行符，就能得到比较满意的结果：

Мы
за
счет
громадного
количества
освобождаемых
должностей
офицеров
можем
в
разы
увеличить
заработную
плату
и
денежное
содержание
гражданского
персонала
и
офицеров
.

但是，俄语的句号是歧义的，文本中的很多圆号并不表示句子结束。例如：1）缩略语： тыс.、млн.、т.е.)；2）时间，如：24.08.2010；3）姓名，如：H.E. Макарова；4)电子邮件或者网络地址，如 mail.ru。俄语中短划线“-”也是有歧义的。必须这些歧义现象进行适当的处理，才能得到真正实用的的俄语分词和断句工具。

5 结论

俄语的自动处理问题是俄语学者在新世纪面临的新问题。这一问题的解决具有重要理论价值和应用价值，是成功解决俄语语料库语言学、文本定量分析的基础，也是构建俄语辅助教学、辅助词典编纂系统的基础。一些基础俄语处理系统，如分词、断句、词形还原与生成、词法分析、句法分析任务急待解决。但是，俄语自动处理的基础和资源还非常薄弱。完成这类工作需要既懂俄语，也懂程序设计语言，甚至要深入到统计语言模型、机器学习的理论中去。我们认为，凡是有志进入俄语计算语言学的深水区的学者，必须掌握至少一门程序设计

语言，Java 则是入门门槛相对不高、使用价值多的程度语言。

附注

1 非俄语词汇

2 http://www.schwebke.com/index.php/10/18/

参考文献

[1]Jeffrey E.F. Friedl. *Mastering Regular Expressions*. 3nd ed[M]. CA.: O'REILY. 2006.

[2]Michael Hammond. *Programming for Linguists: Java Technology for Language Researchers*[M]. Oxford ; Malden, MA : Blackwell Publishers. 2002.

[3]Raymond Gallardo, Scott Hommel, etc. *The Java Tutorial: A Short Course on the Basics*, 6ed[M]. Addison Wesley Proffesional. 2006.

[4]Ben Forta. 正则表达式必知必会[M]. 杨涛译. 北京：人民邮电出版社，2007.

[5]何淑琴. 谈英语文体的定量分析[J]. 外语研究，2005(1).

[6]唐大仕. Java 程序设计[M]. 北京：清华大学出版社，2003.

On Automatic Processing of Russian Texts Using Java Language

Xu Han-cheng

(National University of Defense Technology College of International Relations, Nanjing, China 210039)

Abstract: The paper demonstrates the power of Java language in automatic processing of Russian texts, focusing on the application of Java String class and its regular expression module. The solution is recommended in view of addressing difficulties in input and output of mixed Russian and Chinese texts on Chinese platforms.

Keywords: Russian; NLP; Java; String class; regular expression

作者简介：许汉成，南京国际关系学院博士，教授，主要研究方向：俄语语言学，计算语言学。

收稿日期：2017-11-11 **[责任编辑：叶其松]**