Analysis of Language Resources

Seventh Lecture: Hiroyuki Akama

Noise Words

1) Make a general “stop list” including all the standard noise words
2) Erase the N most frequent words from a particular document

Noise words in English : According to noiseword.txt made by Akasegawa for his application “txtna”,

the to after before so and of in you is that on be…..

Here we treat 3 examples of Perl programming to remove noise words from a target text.

I. Example of the Perl script used to erase the noise words

（The noise words are already included in this script.）

```
-------------------------------Cut here-----------------------------------
#!/bin/perl
#%chmod +x mydelnoise
#%./mydelnoise < hoge.txt >hoge0noise.txt
#hoge.txt: target text; hoge0noise.txt: output.
#!/bin/perl
while(<STDIN>){
s/a¥n//gi;
s/the¥n//gi;
#.....
#The words ”a” and “the” will disappear.
if(/^¥n/){chomp;}
print;}
-------------------------------Cut here-----------------------------------
```

II. How to create the script for erasing the noise words (from a stop list)

```
----------Cut here----------
#!/bin/perl
#################################################################################
#
#Usage:
#%chmod +x delcreate
#%./delcreate < noiseword.txt >delnoise
#%chmod +x delnoise
#%./delnoise < hoge.txt >hoge0noise.txt
#################################################################################
#
print("#!/bin/perl¥n");
# Pay attention to the path of the Perl program.
print("while(<STDIN>){¥n");
while(<STDIN>){
s/(¥S+)/¥s¥/$1¥¥n¥/¥/¥g¥i¥;/g;
#The word ”the” will be transposed into the string “s/the¥n//gi;”.
print;
}
print("if(/^¥¥n/){chomp;}¥n");
print("print;}¥n");
# This is a script to create a script.
----------Cut here----------
```

III How to automatically erase the N most frequent words from a particular document

```
----------Cut here----------
#!/bin/perl
#filename : noiseordercut
#################################################################################
#There will be multiple words with the same frequency, so that there will be some intervals among the values in ordering.
#Let a part of frequency list be
#
```

#231
#220
#103
#103
#100
#100
#100
#98
#
#Then the table for the data will be like this.
#line number(i)rank(j) frequency(f)
#0 1 231
#1 2 220
#2 3 103
#3 3 103
#4 5 100
#5 5 100
#6 5 100
#7 8 98
#8 9 88
#9 9 88
#10 9 88
#11 9 88
#12 13 21

#When the value of j becomes larger than the value of n that is given as argument for the limit rank of the high frequency, **break.**
#if n=7, **break** when j>=7, or i=7
#if n=10, **break** when j>=13, or i=12
#
#usage:% ./noiseorder <filename> <noiselimit> <filewithoutnoise>
#The first argument: file name of the target text
#The second argument: limit of the frequency order to cut noise words
#The third argument: file name of the output without noise.
#We use here the Unix shell inside the “system” function.
##

```
$argn=($file,$limit,$file0noise)=@ARGV;
system("sort $file | uniq -c | sort -r >freqlist");
open(IN, "<freqlist");
$p=0;
while(<IN>){
chomp;
s/^¥s+//g;
#There must be some ¥s to be removed at the top of each line in $freqlist.
@wordinfo=split(/¥s/);
$tmp0[$p]=$wordinfo[0];
$tmp1[$p]=$wordinfo[1];
++$p;
}
close IN;

$i=0;
$j=1;
$k=0;
open(OUT, ">nwlst");
while($j<=$limit){
if($tmp0[$i+1]==$tmp0[$i]){
$j=$k+1;
print OUT $tmp1[$i];
print OUT "¥n";
}
else{
$k=$i+1;
$j=$k+1;
print OUT $tmp1[$i];
print OUT "¥n";
}
++$i;
}
close OUT;
#system("rm freqlist");
open(IN1, "<nwlst");
```

```
open(OUT1, ">delnw");
print OUT1 "#!/bin/perl¥n";
print OUT1 "open(IN2, ¥"<$file¥");¥n";
print OUT1 "while(<IN2>){¥n";
while(<IN1>){
s/(¥S+)/¥s¥/$1¥¥n¥/¥/¥g¥i¥;/g;
print OUT1;
}
print OUT1 "if(/^¥¥n/){¥n";
print OUT1 "chomp;¥n";
print OUT1 "}¥n";
print OUT1 "print;¥n";
print OUT1 "}¥n";
print OUT1 "close IN2;¥n";
close IN1;
close OUT1;
system("chmod +x delnw");
system("./delnw > $file0noise");
# END
```