

# Gender Differences: Polite Terms

Joseph J. Devney

Conventional wisdom in Sociolinguistics has been that women use more polite forms of speech than men do. They apologize more. They say “please” and “thank you” more often. I wanted to find out if this generalization was true in the two corpora available to me, the recorded speech of men (`male.txt`) and women (`female.txt`). To get this information, I wrote a Python script to identify and count the polite constructions, and then calculate ratios for each term in each file.

## Research Questions

The first question to answer is, “What is a polite construction?” The three terms alluded to in the previous paragraph—“please,” “thank” and “sorry”—are only a starting point. For example, not every use of “please” is a polite request. Some uses may indicate sarcasm or some other tone that is not polite. “Britney Spears wants to be on the cover *again*? Oh, *please*.” Saying “I am pleased” or “She will be pleased” is also something other than a request. So a simple search for the string “please” “thank” or “sorry” will find too many instances, but it would be a good first step. The second step would be to discount constructions that could be construed a different way.

Uses of “please” that should not be counted include the following:

- oh please (used sarcastically)
- pleases (as in “he does what he pleases”)
- pleased (as in “she will be pleased”)
- you please (as in “do as you please”)

Similarly, the following variations on “thank you” should be excluded from the analysis:

- thank me (as in “you can thank me for that”)
- get no thanks (as in “I get no thanks for what I do”)
- thanksgiving (the holiday or the type of prayer)
- thanks to (as in “thanks to a traffic jam, I got to work late”)

Finally, the following variations on “sorry” should also be excluded:

- you sorry (as in “aren’t you sorry you asked”)
- be sorry (as in “you’ll be sorry”)

- sorry ass (as in “get your sorry ass out of here”)

## Automating the Data-gathering

The program, called `politetest.py`, was written to extract data from each corpus in two passes for each category of polite construction, and then to perform calculations on the data in order to answer the research questions. The first pass simply searches for occurrences of the basic term, such as “thank,” and totals the number of occurrences. The second pass counts occurrences of phrases that include the basic term but should *not* be included in the total, such as “thank me.” The program also counts the number of words in each corpus. These figures are used in the calculations to determine the rates at which the terms are used in the corpora.

To get the net numbers of apologies, requests, and thanks, the program subtracts the total of exceptions from the total count of the basic term. The results of these calculations are then compared to the total number of words in each corpus. (This total includes the start- and end-line markers. This will increase the total word count, but should not affect the final ratios much.)

To make the data easier to read, the ratios are given in occurrences per ten thousand words, and blank lines separate each set of statistics. In addition, final calculations before printing compare the ratios already calculated. These ratios contain redundant information: the rate for the first file is compared to the rate for the second file, and then the rate for the second file is compared to the rate for the first file. This extra step gives the researcher two different ways to look at the data without any additional effort.

## Results and Conclusions

Using the program `politetest.py` on the files `male.txt` and `female.txt` produced the following results:

Rate of polite terms per ten thousand words:

```
Requests in male.txt is 0.0344451532366.  
Requests in female.txt is 0.0620996066331.  
Ratio male.txt to female.txt rates is 0.554675868402.  
Ratio female.txt to male.txt rates is 1.80285470663.
```

```
Thanks in male.txt is 0.153591502957.  
Thanks in female.txt is 0.310876689303.  
Ratio male.txt to female.txt rates is 0.494059246774.  
Ratio female.txt to male.txt rates is 2.02404874826.
```

```
Apologies in male.txt is 0.435928824568.  
Apologies in female.txt is 0.429017404361.  
Ratio male.txt to female.txt rates is 1.01610988304  
Ratio female.txt to male.txt rates is 0.984145530607.
```

The results for requests (“please” and its variations) and thanks (“thank you” and its variations) confirm the notion that women use more polite formulations than men do. In both cases, women use the polite form approximately twice as often as men do. (Specifically, 1.803 times as often for requests, 2.024 times as often for thanks.)

The narrower version of the “women are more polite” generalization is that women apologize more often than men do. This proposition, however, is not borne out by the data obtained from these two corpora using the criteria identified earlier in this paper. The data show that the men and women whose speech is captured in these corpora apologize at nearly equal rates, with men apologizing slightly more often. Men used “I’m sorry” and its variations 1.016 times as often as women. It seems unlikely that such a marked deviation from both the expected result and the result for the other two terms is due to chance. It may be that men really do apologize in conversation at roughly the same rate as women. Or it may be that the criteria for not counting “sorry” as an apology need to be expanded.

### **Applicability of the Program**

The program `politetest.py` can be used to compare other corpora, to determine if different groups vary in their use of polite constructions. Factors that might be explored include differences in age, in country of origin, in level of education, and in religious background, among others. The constraints on using this program with other corpora are the following:

- Only two corpora at a time can be compared.
- The corpora must be in the form of text files readable by Python.
- The speech transcribed must be in colloquial English.
- The transcription must be in all lower-case letters.
- If the language used by the speakers is other than American English, the lists of strings not counted may need to be modified.

To use `politetest.py` with other corpora, all that needs to be done is to enter the two filenames (with full paths, if necessary) for the corpora to be compared as arguments following the name of the program at the UNIX prompt.

## Appendix: Transcript of politetest.py

```
#!/usr/bin/python
#At the command line, user will enter the name of this program and two
  filenames.
#Import Python modules needed.
import os
import re
import sys
#Create variables with the values of the two filenames entered by the
  user.
file1 = sys.argv[ 1]
file2 = sys.argv[ 2]
#Check that the two filenames entered by the user are valid.
if os.path.isfile(file1) and os.path.isfile(file2):
  #Obtain wordcounts for both input files.
  fileObject = open(file1)
  file1Contents = file.read(fileObject)
  totalcount1 = len(file1Contents)
  fileObject = open(file2)
  file2Contents = file.read(fileObject)
  totalcount2 = len(file2Contents)
#Count occurrences of polite terms in each file.
  pleasepos1 = len(re.findall('please',file1Contents))
  pleasepos2 = len(re.findall('please',file2Contents))
  thankpos1 = len(re.findall('thank',file1Contents))
  thankpos2 = len(re.findall('thank',file2Contents))
  sorrypos1 = len(re.findall('sorry',file1Contents))
  sorrypos2 = len(re.findall('sorry',file2Contents))
#Count occurrences of elements that should not be included in polite
  terms.
  pleaseneg1 = len(re.findall('(pleases | pleased | you please | oh
    please)', file1Contents))
  pleaseneg2 = len(re.findall('(pleases | pleased | you please | oh
    please)', file2Contents))
  thankneg1 = len(re.findall('(thank me | get no thanks | thanksgiv-
    ing | thanks to)', file1Contents))
  thankneg2 = len(re.findall('(thank me | get no thanks | thanksgiv-
    ing | thanks to)', file2Contents))
  sorryneg1 = len(re.findall('(you sorry | be sorry | sorry ass)',
    file1Contents))
  sorryneg2 = len(re.findall('(you sorry | be sorry | sorry ass)',
    file2Contents))
#Calculate actual polite phrases.
  pleasenet1 = pleasepos1 - pleaseneg1
  pleasenet2 = pleasepos2 - pleaseneg2
  thanknet1 = thankpos1 - thankneg1
  thanknet2 = thankpos2 - thankneg2
  sorrynet1 = sorrypos1 - sorryneg1
  sorrynet2 = sorrypos2 - sorryneg2
#Calculate ratios.
  pleaseratio1 = float(pleasenet1) / float(totalcount1)
  pleaseratio2 = float(pleasenet2) / float(totalcount2)
  thankratio1 = float(thanknet1) / float(totalcount1)
  thankratio2 = float(thanknet2) / float(totalcount2)
  sorryratio1 = float(sorrynet1) / float(totalcount1)
  sorryratio2 = float(sorrynet2) / float(totalcount2)
```

```
#Print results.
print "Rate of polite terms per ten thousand words:\n"
print "Requests in %s is %s."%(file1, pleaseratio1 * 10000)
print "Requests in %s is %s."%(file2, pleaseratio2 * 10000)
print "Ratio %s to %s rates is %s."%(file1, file2, pleaseratio1 /
    pleaseratio2)
print "Ratio %s to %s rates is %s.\n"%(file2, file1, pleaseratio2 /
    pleaseratio1)
print "Thanks in %s is %s."%(file1, thankratio1 * 10000)
print "Thanks in %s is %s."%(file2, thankratio2 * 10000)
print "Ratio %s to %s rates is %s."%(file1, file2, (thankratio1 /
    thankratio2))
print "Ratio %s to %s rates is %s.\n"%(file2, file1, (thankratio2 /
    thankratio1))
print "Apologies in %s is %s."%(file1, sorryratio1 * 10000)
print "Apologies in %s is %s."%(file2, sorryratio2 * 10000)
print "Ratio %s to %s rates is %s"%(file1, file2, (sorryratio1 /
    sorryratio2))
print "Ratio %s to %s rates is %s.\n"%(file2, file1, (sorryratio2 /
    sorryratio1))
```