Document Retrieval using TF-IDF Weighted Rank and TF-IDF Cosine Similarity

In [1]:

```
pip install num2words
```

```
Requirement already satisfied: num2words in c:\python39\lib\site-packages (0.5.10)
Requirement already satisfied: docopt>=0.6.2 in c:\python39\lib\site-packages (from
num2words) (0.6.2)
Note: you may need to restart the kernel to use updated packages.
WARNING: You are using pip version 21.3.1; however, version 22.0.4 is available.
You should consider upgrading via the 'C:\Python39\python.exe -m pip install --upgr
ade pip' command.
```

Imports

In [2]:

```
from zipfile import ZipFile
zf = ZipFile('stories.zip', 'r')
zf.extractall()
zf.close()
```

In [3]:

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from collections import Counter
from num2words import num2words

import nltk
import os
import string
import numpy as np
import copy
import pandas as pd
import pickle
import re
import math

# %load_ext autotime
```

In [4]:

```
title = "stories"
alpha = 0.3
```

Taking all folders

In [5]:

```
folders = [x[0] for x in os.walk(str(os.getcwd())+'/'+title+'/')]
folders[0] = folders[0][:len(folders[0])-1]
```

In [6]:

```
folders
```

Out[6]:

```
['C:\\Users\\user/stories',
 'C:\\Users\\user/stories/FARNON',
 'C:\\Users\\user/stories/SRE']
```

Collecting the file names and titles

In [7]:

```python
dataset = []

c = False

for i in folders:
    file = open(i+"/index.html", 'r')
    text = file.read().strip()
    file.close()

    file_name = re.findall('><A HREF="(.*)">', text)
    file_title = re.findall('<BR><TD> (.*)\n', text)

    if c == False:
        file_name = file_name[2:]
        c = True

    print(len(file_name), len(file_title))

    for j in range(len(file_name)):
        dataset.append((str(i) +"/"+ str(file_name[j]), file_title[j]))
```

```
452 452
0 0
15 15
```

In [8]:

```python
len(dataset)
```

Out[8]:

```
467
```

In [9]:

```python
N = len (dataset)
```

In [10]:

```python
def print_doc(id):
    print(dataset[id])
    file = open(dataset[id][0], 'r', encoding='cp1250')
    text = file.read().strip()
    file.close()
    print(text)
```

Preprocessing

In [11]:

```python
def convert_lower_case(data):
    return np.char.lower(data)
```

In [12]:

```
def remove_stop_words(data):
    stop_words = stopwords.words('english')
    words = word_tokenize(str(data))
    new_text = ""
    for w in words:
        if w not in stop_words and len(w) > 1:
            new_text = new_text + " " + w
    return new_text
```

In [13]:

```
def remove_punctuation(data):
    symbols = "!\"#$%&()*+-./:;<=>?@[\]^_`{|}~\n"
    for i in range(len(symbols)):
        data = np.char.replace(data, symbols[i], ' ')
        data = np.char.replace(data, "  ", " ")
    data = np.char.replace(data, ',', '')
    return data
```

In [14]:

```
def remove_apostrophe(data):
    return np.char.replace(data, "'", "")
```

In [15]:

```
def stemming(data):
    stemmer= PorterStemmer()

    tokens = word_tokenize(str(data))
    new_text = ""
    for w in tokens:
        new_text = new_text + " " + stemmer.stem(w)
    return new_text
```

In [16]:

```
def convert_numbers(data):
    tokens = word_tokenize(str(data))
    new_text = ""
    for w in tokens:
        try:
            w = num2words(int(w))
        except:
            a = 0
        new_text = new_text + " " + w
    new_text = np.char.replace(new_text, "-", " ")
    return new_text
```

In [17]:

```
def preprocess(data):
    data = convert_lower_case(data)
    data = remove_punctuation(data) #remove comma seperately
    data = remove_apostrophe(data)
    data = remove_stop_words(data)
    data = convert_numbers(data)
    data = stemming(data)
    data = remove_punctuation(data)
    data = convert_numbers(data)
    data = stemming(data) #needed again as we need to stem the words
    data = remove_punctuation(data) #needed again as num2word is giving few hypens
    data = remove_stop_words(data) #needed again as num2word is giving stop words 1
    return data
```

Extracting Data

In [18]:
```
processed_text = []
processed_title = []

for i in dataset[:N]:
    file = open(i[0], 'r', encoding="utf8", errors='ignore')
    text = file.read().strip()
    file.close()

    processed_text.append(word_tokenize(str(preprocess(text))))
    processed_title.append(word_tokenize(str(preprocess(i[1]))))
```

Calculating DF for all words

In [19]:
```
DF = {}

for i in range(N):
    tokens = processed_text[i]
    for w in tokens:
        try:
            DF[w].add(i)
        except:
            DF[w] = {i}

    tokens = processed_title[i]
    for w in tokens:
        try:
            DF[w].add(i)
        except:
            DF[w] = {i}
for i in DF:
    DF[i] = len(DF[i])
```

In [20]:
```
# DF
```

In [21]:
```
total_vocab_size = len(DF)
```

In [22]:
```
total_vocab_size
```

Out[22]:
```
32350
```

In [23]:
```
total_vocab = [x for x in DF]
```

In [24]:
```
print(total_vocab[:20])
```

```
['sharewar', 'trial', 'project', 'freewar', 'need', 'support', 'continu', 'one', 'h
undr', 'west', 'fifti', 'three', 'north', 'jim', 'prentic', 'copyright', 'thousand
', 'nine', 'nineti', 'brandon']
```

In [25]:

```
def doc_freq(word):
    c = 0
    try:
        c = DF[word]
    except:
        pass
    return c
```

Calculating TF-IDF for body, we will consider this as the actual tf-idf as we will add the title weight to this.

In [26]:

```
doc = 0

tf_idf = {}

for i in range(N):

    tokens = processed_text[i]

    counter = Counter(tokens + processed_title[i])
    words_count = len(tokens + processed_title[i])

    for token in np.unique(tokens):

        tf = counter[token]/words_count
        df = doc_freq(token)
        idf = np.log((N+1)/(df+1))

        tf_idf[doc, token] = tf*idf

    doc += 1
```

In [27]:

```
# tf_idf
```

Calculating TF-IDF for Title

In [28]:
```
doc = 0

tf_idf_title = {}

for i in range(N):

    tokens = processed_title[i]
    counter = Counter(tokens + processed_text[i])
    words_count = len(tokens + processed_text[i])

    for token in np.unique(tokens):

        tf = counter[token]/words_count
        df = doc_freq(token)
        idf = np.log((N+1)/(df+1)) #numerator is added 1 to avoid negative values

        tf_idf_title[doc, token] = tf*idf

    doc += 1
```

In [29]:
```
# tf_idf_title
```

In [30]:
```
tf_idf[(0,"go")]
```

Out[30]:
```
0.0002906893990853149
```

In [31]:
```
tf_idf_title[(0,"go")]
```

Out[31]:
```
0.0002906893990853149
```

Merging the TF-IDF according to weights

In [32]:
```
for i in tf_idf:
    tf_idf[i] *= alpha
```

In [33]:
```
for i in tf_idf_title:
    tf_idf[i] = tf_idf_title[i]
```

In [34]:
```
len(tf_idf)
```

Out[34]:
```
344378
```

TF-IDF Matching Score Ranking

In [35]:

```
def matching_score(k, query):
    preprocessed_query = preprocess(query)
    tokens = word_tokenize(str(preprocessed_query))

    print("Matching Score")
    print("\nQuery:", query)
    print("")
    print(tokens)

    query_weights = {}

    for key in tf_idf:

        if key[1] in tokens:
            try:
                query_weights[key[0]] += tf_idf[key]
            except:
                query_weights[key[0]] = tf_idf[key]

    query_weights = sorted(query_weights.items(), key=lambda x: x[1], reverse=True)

    print("")

    l = []

    for i in query_weights[:10]:
        l.append(i[0])

    print(l)


matching_score(10, "Without the drive of Rebeccah's insistence, Kate lost her momen
```

```
Matching Score

Query: Without the drive of Rebeccah's insistence, Kate lost her momentum. She stoo
d next a slatted oak bench, canisters still clutched, surveying

['without', 'drive', 'rebeccah', 'insist', 'kate', 'lost', 'momentum', 'stood', 'ne
xt', 'slat', 'oak', 'bench', 'canist', 'still', 'clutch', 'survey']

[166, 200, 352, 433, 211, 350, 175, 187, 188, 294]
```

In [36]:

```
print_doc(2)
```

```
('C:\\Users\\user/stories/14.lws', 'A Smart Bomb with a Language Parser')
----------------------------------------------
"The Adventures of Lone Wolf Scientific"
------------------------------------------
An electronically syndicated series that
follows the exploits of two madcap
mavens of high-technology. Copyright 1991
Michy Peshota. All rights reserved.  May
not be distributed without accompanying
WELCOME.LWS and EPISOD.LWS files.
-----------------------
EPISODE #14


           A Smart Bomb with a Language Parser

>>>S-max attempts to thwart The Last Words Bomb's language
parser, but to no avail.  He discovers that program code is
often more stubborn than human will.<<

                      By M. Peshota

     "Whoever heard of a smart bomb with a language parser?"
he heard him grumble.  Austin watched his wild-haired
officemate, his bull-like features creased into a scowl,
hunched over stacks and stacks of thesauruses, whipping
their pages, cursing bitterly.  "Only a nudnik programmer
would think of making a bomb verbally context-sensitive," he
growled.

     Earlier in the evening, the computer builder had come
to him, his condescending eyes moist with humility, his
normally Napoleanic upper lip quivering helplessly, and
begged the hollow-eyed wizard to recode Andrew.BAS's guided
missile software.  Specifically, he wanted him to recode it
so that the computer would not screech alarms and its screen
flash bright red whenever he keyed in at its screen prompt
the declaration "Gus Farwick is a testosterone-less simp
with eel toes for brains!"  But, as much as the assembly
language savant would have liked to become involved in such
a worthwhile project, he was too preoccupied at the moment
with his many neurotic frets, especially his fear of the
possible return of the ghost of Alan Turing to his former
domicile in Austin's office coat closet, to be able to do
anything but gape zombie-like into the flourescent-white
night air and drool down the front of his checked shirt
until eventually the computer builder shuffled away.

     Still gaping, Austin could hear him pawing through the
section of the thesaurus that listed synonyms for
"testosterone-less simp."  "Ninnyhead.  Puddingbrain.
Knucklenoggin," he recited in his nasal drone.  He
laboriously typed them one by one into the guided missile
software, then groaned as the screen flashed red in response
and the alarm bells chimed.  "This is what I get for having
familiarized that twit programmer with my entire range of
verbal invective," he grunted, flinging open another
thesaurus.  He raised his head and mused, "Maybe if I tried
some alternate spellings...."  After some thought, he typed
```

```
into the machine "Gees...Farwoook...is...a...
Tusktossturoon-Mess Imp...Wif...Eeeel-Tooeys...4...Brains!"
The computer responded with a long, slow gag, then flashed
its screen red and chimed like a maimed pinball machine.
The computer builder slammed his fist on the desk in rage.

     Tired, the assembly language savant nestled his head on
the worn ivories of his keyboard and listened to his
officemate's wild, futile linguistic manipulations until
late in the night.  Eventually he fell asleep.  In his
troubled dreams, he thought he saw the flyblown profile of
the ghost who dogged him, who terrified him day and night
with his incessant ravings about long-forgotten computer
memory registers, the irrepressible ghost of Alan Turing,
the father of programming.  Turing materialized, tweed suit,
shabby wingtips, cobwebbed copy of <<Byte>>, battered
bicycle and all, in back of the computer builder's zebra-fur
cloaked chair.  With a devil-may-care glower that was not
unlike the computer builder's own condescending smirk, he
extended shadowy hands over the latter's shoulder.  He took
hold of the computer builder's Hanswurst knuckles, and, with
the impassioned vigor of a symphonic conductor, guided them
into a manic dance across the terminal's keys.

     The computer builder, unaware of the ghost's presence,
watched his gamboling hands, aghast.  When his finger were
finally still, lying in an artistically spent, twisted heap,
like the hands of Beethoven on the numeric keypad, he looked
at them in surpise, then glanced up at the screen.
"Omigod!" he gasped.  "I have done it!  My genius has won
out again!  I have found a way to disable Andrew.BAS's kooky
language parser!"  He smiled with pride at the string of
inscrutable algorithms marching across the screen.  "Gawd,
how I wish I could understand what those are," he clucked,
typing into the missile software "testosterone-less simp,"
adding "with eel toes for brains."  He pressed 'enter' and
listened closely, but heard no warning bells, nor did he see
the screen flash red.  He smiled, "Gus Farwick, prepare to
read your epitaph in the sky."  The ghost nodded with
approval, got on his bike, and disappeared, and the computer
builder leaped from his chair and hopped from foot to foot
like a wound up harlequin.


                          <Finis>


>>Is trouble on the way when Dingready & Derringdo Aerospace
demonstrates their latest crop of computer-guided weapons to
military nabobs?  Find out in the next episode of 'The
Adventures of Lone Wolf Scientific.'<<
```

TF-IDF Cosine Similarity Ranking

In [37]:

```
def cosine_sim(a, b):
    cos_sim = np.dot(a, b)/(np.linalg.norm(a)*np.linalg.norm(b))
    return cos_sim
```

Vectorising tf-idf

In [38]:

```
D = np.zeros((N, total_vocab_size))
for i in tf_idf:
    try:
        ind = total_vocab.index(i[1])
        D[i[0]][ind] = tf_idf[i]
    except:
        pass
```

In [39]:

```
def gen_vector(tokens):

    Q = np.zeros((len(total_vocab)))

    counter = Counter(tokens)
    words_count = len(tokens)

    query_weights = {}

    for token in np.unique(tokens):

        tf = counter[token]/words_count
        df = doc_freq(token)
        idf = math.log((N+1)/(df+1))

        try:
            ind = total_vocab.index(token)
            Q[ind] = tf*idf
        except:
            pass
    return Q
```

In [40]:

```
def cosine_similarity(k, query):
    print("Cosine Similarity")
    preprocessed_query = preprocess(query)
    tokens = word_tokenize(str(preprocessed_query))

    print("\nQuery:", query)
    print("")
    print(tokens)

    d_cosines = []

    query_vector = gen_vector(tokens)

    for d in D:
        d_cosines.append(cosine_sim(query_vector, d))

    out = np.array(d_cosines).argsort()[-k:][::-1]

    print("")

    print(out)

#     for i in out:
#         print(i, dataset[i][0])

Q = cosine_similarity(10, "Without the drive of Rebeccah's insistence, Kate lost he
```

```
Cosine Similarity

Query: Without the drive of Rebeccah's insistence, Kate lost her momentum. She stoo
d next a slatted oak bench, canisters still clutched, surveying

['without', 'drive', 'rebeccah', 'insist', 'kate', 'lost', 'momentum', 'stood', 'ne
xt', 'slat', 'oak', 'bench', 'canist', 'still', 'clutch', 'survey']

[200 166 433 175 169 402 211  87 151 369]
```

In [41]:

```
print_doc(200)
```

```
('C:\\Users\\user/stories/ghost', 'Time for Flowers, by Gay Bost')
TIME FOR FLOWERS
  by Gay Bost

They'd put flowers up. She hadn't noticed. Time wouldn't hold still.
She remembered, quite clearly, that time had been a simple thing; one
moment following the previous one, seconds strung out neatly like her
mother's pearls laid out on the dark mahogany vanity each Sunday
morning. But there had been a catch . . .

Hung around Mother's neck the catch clicked and the tidy little line
of seconds became a never ending circle with only the catch in the
middle. For some reason the thought of pearls gathered from the sea,
naturally nested within the confines of oyster shells, scattered
haphazardly about the ocean floor disturbed her.

Now they'd put up the flowers in the same careless groupings. This,
too, disturbed her. Bright yellow trumpets, their collars spread to
catch the sun, dotted the front yard in clusters of two or three, five
or six. Bunches laid carelessly and forgotten. In a moment she'd
come away from the window and have a word with the gardener. He
listened so well and explained to others so reasonably why this should
be so instead of the way they wanted it done, how that would look
better or cut the wind more effectively.

And then she recalled his stiff body stretched out in the little bed
over the garages. Another pearl had come loose from the strand,
seeming to want to search out its old home in a far away oyster bed.
She would have those pearls laid out neatly, one following the one
before and so on and so on. She would have those damned yellow
flowers marching smartly along the walk. She'd have it if she
had to go out there and replant each and every one of them.

She flew down the hallway and sailed over the steps leading the
back way to the kitchen, much as she had done as a child. Where then
she had skipped in joy she now catapulted her form in anger.

"And there you are!" she said, as she encountered the woman she had
come to know as Kate. All of five foot tall in her stocking feet and
surely every bit of two hundred pounds, her pudgy fists more often
than not braced on the sudden outburst of her hips. So she stood,
having turned from the sink. Suds and water darkened the fabric of her
dress. Her face was pleasant; round, rosy cheeked, with eyes the color
of mint in the summer sunset. "And *where have you been these three days*?"

"I want the flowers straightened out," Rebeccah said. "I want the
flowers placed in the proper alignments."

Kate tilted her head, narrowed her eyes and frowned. "Ah, you're in a
huff again. What can it be this time?"

"I want the flours straightened out," Rebeccah yelled, coming up to
the woman's face.

Kate went directly to the cupboard, strained upon her tiny toes to
reach the second shelf, and pulled the flour canister out. She set it
on the counter. She repeated the process, bringing out a smaller
canister. Rebecca knew this one to be the unbleached flour Kate used
for one particular recipe.
```

```
"No,no, no!"  Rebeccah hissed. "Flowers!  Not flours!"  She propped
herself against the edge of the kitchen table and crossed her arms
over her chest, waiting for the woman to get it right.

Kate stood looking dumbly at the canisters. "Now, what was I going to
do with these?"  she asked herself. She drummed her fingers on the
counter top before bringing one hand to her lips, where the pointer
finger tapped on her upper lip.

"The Flowers!  Outside!"  Rebecca screamed, highly agitated.

Kate gathered the two canisters and moved toward the back door, one
held against her ample form by each arm.

Exasperated, Rebeccah followed her out, watching to see what she would do.

Without the drive of Rebeccah's insistence, Kate lost her momentum.
She stood next a slatted oak bench, canisters still clutched, surveying
the sunlit yard and gardens beyond. Harold had done a passable job
trimming the hedges, but Kate missed the gardener's touch. She resolved
to contact the nursery and find another. Flaux, bright purples, pinks
and radiant white encircled the herb garden, a brilliant contrast to
the varied greens within. She set the canisters down on the bench and
moved toward the cheerful scene.

Rebeccah, discouraged, sat primly on the edge of the bench, dusting a
wisp of hair away from her temple. New mint, dew draped, veiled a
border of stocky wooden poles to trail onto the walk, had been crushed,
probably by the man of the house on his way off to work. The scent
filled her nostrils. She found herself a child, again, tasting her
first tea with mint -- fresh cut from the gardens. _"How long has it
been?"_  she wondered. Kate had gone down on her knees over the flaux,
bending to weed through the thyme.

"I don't know why I have to put up with idiots," Rebeccah complained.
"It all so worthless, so futile."  With a great sigh she rose from the
bench and made her way back into the house. The bright kitchen seemed
a waste of life, all a travesty to cover the desolation of her
unnaturally extended existence.

She faced the stairs with exhaustion, deciding, instead, to forego the
trip up. She sat on the bottom step, delicate chin propped on tightly
curled fists, gazing dully at the open pantry door, seeing into the past
-- again. Where, in this world the shelves were haphazardly stacked with
cans of peaches and corn, she saw row after row of glass jars. Beets!
Ugh!  Her grandmother's pickled beets, always pretty to view, left a
phantom bitterness within her mouth.

On the lawn Kate sat back on her heels, suddenly lost in sorrow and
self-pity. Tears streamed down her cheeks to drop onto the fabric of
her dress. She thought of Harold, busily showing homes as lovely as
their own to strangers while she ruined her nails weeding this pitiful
excuse for a garden. She shoved her pudgy fists into her burning eyes
and wept aloud for the waste of her life. She sniffed back her running
nose . . . sniffed again. She snuffled like a dog scenting something
unusual, nose in the air. "Beets?"  she asked aloud. "Beets?"  Her
hands dropped to her thighs, pushing to rise. _"Of course,"_ she thought
to herself, _"this *lovely* house is haunted by a very emotional woman."_
```

```
Her knees ached. She turned toward the house and noticed the flour
canisters on the bench. "And whatever she wants *this* time is not
getting through this thick skull of mine!"

Kate knuckle-rapped herself above her right temple. "Rebeccah!"  she
called. "Quit moping!  You'll ruin another day for me and I still
have to deal with that horrible Avon woman this morning."

"I want my flowers properly aligned!" Rebeccah screamed from the stairs.

As Kate passed the bench she paused to move the flour canisters so
that the labels faced in the same direction, each perfectly centered
over three of the wood slats. With a self-satisfied air she re-entered
her own kitchen. "Now," she began, addressing the refrigerator, "what
we need is improved communication."

"Fool," hissed Rebeccah, "you're talking to the refrigerator again."

"You don't want an empath. You want a telepath," Kate said, turning
to stare at Rebeccah with surprising accuracy.

The two women blinked at each other and broke into laughter.

"I want my flowers straightened out!"  Rebeccah commented softly when
the mirth had passed.

                              * * *

"There!"  Kate replaced the telephone hand piece and pocketed the
scrap of paper she'd written the new gardener's name upon. "Mr.
Hi-a-cow-wah," she practiced aloud. "Very good."  The door chime rang
throughout the house, echoing off the tiled kitchen walls.

"Oh, no!"  wailed Rebeccah. "Not Japanese!  They have such spiritual
ideas on gardening -- I'll never get through to him!"

"Oh, dear!"  Kate bemoaned, certain the Avon woman had come to call.
She brushed her hands over her skirt, straightened her broad shoulders
and pushed through to the dining room, determined not to buy a single
thing today.

"Good morning, Mrs. Blanchard!"  beamed the woman in the pale rose
colored ensemble. Purse clutched in one hand, sample case in the other,
she reminded Kate of the Lady Justice, scales perfectly balanced. But
this lady had no blindfold. (All the better to see you with, my dear.
And Oh, wouldn't this color just bring on the blush in your cheeks for
$11.00 a tube?)  "Isn't it just a glorious day?" the woman pronouned,
boldly stepping over the threshold on past assumptions.

_"That's it!"_ Kate thought to herself. She'd let the woman in once,
bought gifts soaps and lipstick in the spirit of cooperation, and
never been free of past assumptions since. "Glorious!" Kate echoed,
moving aside before she was trod upon. Rebeccah hovered at the dining
room doors. Kate felt her there.

"Oh, and you've brought the day in with you!" exclaimed the woman,
noting cut flowers on mantel and coffee table. "How healthful!"

"Healthful?" Kate inquired.
```

```
"Oh, yes. Studies have shown that people who surround themselves with
live plants and fresh flowers indoors live longer, feel better, and
enjoy life more fully."

"Coffee?"  Kate offered as the woman sat on the edge of the sofa. It
was the one torment she allowed herself to use on the woman, knowing
full well this door to door saleswoman would shun other people's
bathrooms.

"No thank you," she answered, a slight grimace flashing across her
face as she scooted forward and opened her case.

"You're so rude!" Rebeccah crowed, having come closer. "She's got a
bladder full now."

Kate smiled, holding back a giggle. She was certain she'd scored
without knowing why. The woman drew forth brightly colored sheets of
paper and placed them neatly before Kate on the glass topped table.
_"A promotional,"_ Kate moaned within her mind. At the bottom of each
was stamped, in flowing script, "Eleanor Thomsason."  Address and two
phone numbers followed in block lettering.

"I don't really need anything today, Eleanor," Kate began.

"Of course you don't, dear. You're more than lovely in your house
frock and clean scrubbed face. But you must see the new complexion
care line we're offering. Designed especially for the woman over 30
and her special needs," Eleanor pulled full sized display item from
the depths of her bottomless case and set them neatly in a row,
labels facing the prospective buyer. "As you can see here," she said
crisply, long manicured finger nail tapping each item gently as she
spoke, "We have a scrub, toner, tightener, moisturizer and light
foundation. The foundation comes in 6 basic colors. Just to smooth
over those tiny blotches we all seem to have after 30."

Kate sat forward in her occasional chair, considering the possibility
that she might, indeed, need a little more complexion care. She
touched the toner, tilting it slightly to the light. While she was
otherwise engaged Eleanor brought forth tubes, bottles and jars of the
same line. She busied herself arranging them in a straight line to
the left and just behind the first row.

"And here we have the corresponding blush, highlighters, lipsticks
and shadows. Now this line is made with completely natural base
substances," Eleanor pointed out.

"Chemicals," Rebeccah commented, coming closer still, intently
interested in the ordered presentation.

Kate let go the toner and reached for the blush. Eleanor straightened
the toner, turning the label toward the prospective buyer. Rebeccah
came around the coffee table and sat on the sofa with Eleanor, her
arms primly at her sides, hands clasped in her lap. Rebeccah leaned
forward in the same manner as did Eleanor.

The genial rise and fall of the woman's voice slipped into the background
of sounds passing by on the peaceful street outside. Kate blinked once,
the blush still clasped within her fingers, watching Eleanor's lips move.
```

```
She could almost hear Rebeccah.

Rebeccah's attention was focused entirely on Eleanor the Avon lady.
"The flowers have been scattered willy-nilly along the walk,"
Rebeccah said conversationally, her lips mere inches from Eleanor's
ear. "They look so untidy."  Eleanor looked, suddenly, as if she'd
forgotten something. Kate remembered the flour canisters on the
bench. "What we need is someone with some organizational ability,"
Rebeccah continued.

Eleanor drew forth her order book. "Flowers are like life's little
markers," Rebeccah whispered. Eleanor reached into her case for a
marker. "Yellow markers, as it were, for the days of our lives."
Eleanor replaced the fine tipped black marker and retrieved a broad
stroke yellow highlighter. Kate seemed to hear McDonald Carey speaking
about sand. "The flowers along the walk NEED straightening."

"Will you excuse me, just one moment?" Kate asked. She knew exactly
where to find that hourglass. She rose from her chair

"Certainly, dear," Eleanor answered, her mind seemingly elsewhere
while her hands compulsively aligned the display items.

"*YOU* could be the only one for the job!"  Rebeccah spoke
authoritatively, her body turned toward Eleanor. "The flowers need
alignment!"

Kate felt an oppressive headache coming on. Two of them in one
morning was more than anyone should be expected to bear. As she
passed through the kitchen door her spirits seemed to rise suddenly.
Sunshine slanted into the room to highlight every gleaming surface,
glinting sweetly on glassware and chrome. She inhaled fully, filling
her lungs with the aroma of fresh brewed coffee. The hourglass
spilling out the days of her life seemed important only in the
abstract. All was right today. She thought of the flowers by the
walk, then. For some reason she  wanted to see them from the top
floor.

She poured herself a cup of coffee, carried it up the back stairs to
the second floor landing and peered from the window into the side
yard. She thought, idly, of the new gardener, and what creative
expression he might come up with for that spot there, which had never
been cultivated. Onward, to the front of the house, and into the
quiet room beneath the pitch of the front eaves.

She sat on the window ledge and balanced her cup on the sill, the
threatened headache a memory, only, of Saturday afternoons with her
mother. Somewhere behind her temples her mother's voice droned on and
on; something about book spines and the edge of the shelf. Sometimes
one had to learn to ignore the librarian in order to read the books.

Her eyes drifted to the front walk. Far below, as if in another
world, Eleanor the Avon lady knelt in the grass next to the walk.
A tall shadow stood near, softly, insistently coaxing, as Eleanor
carefully spaded deep into the earth and removed a daffodil. She
placed it gently into a prepared hole, tamped the earth around it and
proceeded to dig another hole, exactly six inches from the last, in a
perfectly straight line parallel to the walk.
```

```
"Oh, for crying out loud!"  Kate exclaimed, watching closely. "Those
flowers!"  She'd have to remember to collect the flour canisters
before Harold came home. "Goodness, Rebeccah," she continued, with
some exasperation, "why on earth didn't you say `Daffodils'?"

                              # # #

Copyright 1994 Gay Bost
---------------------------------------------------------------------------
 Gay is a Clinical Lab Tech with experience in Veterinary medicine.
Originally from NORTHERN California, she has resided in Southeast Missouri
with her husband and an aggressive 6 year old boy, since 1974. She
installed her first modem in the summer of 1992 and has been exploring new
worlds since. Her first and only publication, a short horror story, came
when she was 17 years old. The success was so overwhelming she called an
end to her writing days and went in search of herself. She's still looking.
You will find Gay's work in the best Electronic Magazines.
============================================================================
```

In [ ]: