```
"""Spelling Corrector.

Copyright 2007 Peter Norvig.

Open source code under MIT license: http://www.opensource.org/licenses/mit-license.php
"""

import re, collections

def words(text): return re.findall('[a-z]+', text.lower())

def train(features):
    model = collections.defaultdict(lambda: 1)
    for f in features:
        model[f] += 1
    return model

NWORDS = train(words(file('big.txt').read()))

alphabet = 'abcdefghijklmnopqrstuvwxyz'

def edits1(word):
   s = [(word[:i], word[i:]) for i in range(len(word) + 1)]
   deletes    = [a + b[1:] for a, b in s if b]
   transposes = [a + b[1] + b[0] + b[2:] for a, b in s if len(b)>1]
   replaces   = [a + c + b[1:] for a, b in s for c in alphabet if b]
   inserts    = [a + c + b     for a, b in s for c in alphabet]
   return set(deletes + transposes + replaces + inserts)

def known_edits2(word):
    return set(e2 for e1 in edits1(word) for e2 in edits1(e1) if e2 in NWORDS)

def known(words): return set(w for w in words if w in NWORDS)

def correct(word):
    candidates = known([word]) or known(edits1(word)) or known_edits2(word) or [word]
    return max(candidates, key=NWORDS.get)

##############
```

Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}.$$

We will say that we are trying to find the correction *c*, out of all possible corrections, that maximizes the probability of *c* given the original word *w*:

argmax$_c$ P(*c*|*w*)

By Bayes' Theorem this is equivalent to:

argmax$_c$ P(*w*|*c*) P(*c*) / P(*w*)

Since P(*w*) is the same for every possible *c*, we can ignore it, giving:

argmax$_c$ P(*w*|*c*) P(*c*)

There are three parts of this expression. From right to left, we have:

1. P(*c*), the probability that a proposed correction *c* stands on its own. This is called the **language model**: think of it as answering the question "how likely is *c* to appear in an English text?" So P("the") would have a relatively high probability, while P("zxzxzxzyyy") would be near zero.
2. P(*w*|*c*), the probability that *w* would be typed in a text when the author meant *c*. This is the **error model**: think of it as answering "how likely is it that the author would type *w* by mistake when *c* was intended?"
3. argmax$_c$, the control mechanism, which says to enumerate all feasible values of *c*, and then choose the one that gives the best combined probability score.