Rijndael

Write a Python program that implements a Rijndael cipher that takes a ciphertext as input from `stdin` and outputs the correct plaintext and corresponding key used in the cipher to `stdout`. You will be provided with a dictionary of words that is **guaranteed** to contain the key used to generate the ciphertext. Note that Rijndael is the cipher that was selected for the Advanced Encryption Standard (AES).

Since you will be playing the part of cryptanalysts, your program will need to generate candidate plaintexts using keys in the provided dictionary. Once candidate plaintexts are generated, you must **efficiently** filter out invalid ones using a method of your choice. As in previous programs, one way is to compare "words" in candidate plaintexts with words in the provided dictionary. Note that you may want to "normalize" words in candidate plaintexts and the dictionary (e.g., remove punctuation, convert to lowercase, etc). If enough words match (say, three-fourths of them), a candidate plaintext becomes likely. You can also consider using the average length of words in English US to help identiy candidate plaintexts.

Notes and Requirements:

- Submit your source code only. I will provide my own ciphertext to test with;
- Read the ciphertext from `stdin`;
- Write the plaintext and key to `stdout`;
- Comment your source code appropriately; and
- The ciphertext could contain multiple lines.

Please, no GUIs. Make this a command line application without frills that I can execute at the command line as illustrated below via several sample runs of my program:

```
jgourd@latech:~$ python Rijndael.py < ciphertext-1.txt
KEY=heartburn
The lady said, "Oh my! You have nice eyes. Are they yours?"
I laughed.

jgourd@latech:~$ python Rijndael.py < ciphertext-2.txt
KEY=Mercury
Never attribute to malice that which is adequately explained by stupidity.
(Hanlon’s razor)
```