How to Tell if a Text is English or Not
Exposition by William Gasarch

1 Why is important to tell A Text Is English

When I asked the question

Is the shift cipher secure?

students often say

No, since there are only 26 shifts and you can test each one!

How would you really DO this? You could have a computer produce SHIFT1(TEXT), SHIFT2(TEXT),...,SHIFT25(TEXT). You could then LOOK at all of them and see which one LOOKS like English. That would work, but take a lot of effort. Can you mechanize the *LOOKS like English* part? Why is this important? For one thing, consider the following question:

Is the affine cipher secure?

Would you say:

No, since there are only 312 possibilities and you can test each one!

How would you really DO this? You could have a computer produce all 312 possible affine's (not sure there is a good word for it) which I will call $U_1, \ldots, U_{312}$. You could then LOOK at all of them and see which one LOOKS like English. That would work, but take A LOT A LOT A LOT of effort. Can you mechanize the *LOOKS like English* part? This is now clearly important since looking at 312 texts is hard (unless you have 312 TAs, but I only have 2).

2 Transforming the Problem Into a Math Problem

Let p_a be the prob that a randomly chosen letter of an English Text is a.

Let p_b be the prob that a randomly chosen letter of an English Text is b.

etc.

These values have been tabulated. They may vary some if you know you are looking at certain types of documents. For examples, if you are looking at documents about the Middle East then q will be more common than it usually is because of Iraq and Qatar. But for now assume that we have a $p_a, \ldots, p_z$ that are approximately correct for all (long enough) documents.

Note that, because these are probabilities of disjoint events, the p's are between 0 and 1 and sum to 1.

Let $\vec{p} = (p_a, p_b, \ldots, p_z)$.

Assume T is a document that you wonder if it is English or not. Perhaps you originally had document S and you think S was coded by the shift cipher, so you are looking at $SHIFT1(S)$, ..., $SHIFT25(S)$. Let T be $SHIFT1(S)$.

Here are the first few steps of what you would do:

1. Go through T and find the frequencies of all of the letters (WRITE A PROGRAM THAT DOES THIS EASILY).
2. Let $f_a, \ldots, f_z$ be those frequencies. Let $F = f_a + \cdots + f_z$.
3. Let $q_a = f_a/F$, $q_b = f_b/F$, ..., $q_z = f_z/F$.
4. Let $\vec{q} = (q_a, \ldots, q_z)$.
5. NOW WE NEED TO SEE IF $\vec{p}$ AND $\vec{q}$ ARE SIMILAR.

WE now have a new problem which is the title of the next section

3 Given Two Vectors of Probabilities How Can You Test If They Are Similar?

Let $\vec{p}$ and $\vec{q}$ be two vectors of probabilities of length n. We discuss several measures of how similar they are
1) Look at $\sum_{i=1}^{n}(p_i - q_i)$.
There is a problem with this: what if
$\vec{p} = (0.2, 0, 0.2, 0, 0.2, 0, 0.2, 0, 0.2, 0)$
and
$\vec{q} = (0, 0.2, 0, 0.2, 0, 0.2, 0, 0.2, 0, 0.2)$
These vectors are NOT similar yet
Then $\sum_{i=1}^{10}(p_i - q_i) = 0.2 - 0.2 + 0.2 - 0.2 + 0.2 - 0.2 + 0.2 - 0.2 + 0.2 - 0.2 = 0$
So how to get around this:
2) Look at $\sum_{i=1}^{n} |p_i - q_i|$.
This does get around the problem above. But absolute value is hard to work with mathematically and there are other things wrong with it that I won't go into here.
3) Lets first do an example. The Vorlons have a 5-letter alphabet a, b, c, d, e (WOW-they use the first five letter of English! What a coincidence!) Assume
$(p_a, p_b, p_c, p_d, p_e) = (0.5, 0.2, 0.1, 0.1, 0.1)$
We have a text which has
$(q_a, q_b, q_c, q_d, q_e) = (0.48, 0.15, 0.13, 0.12, 0.12)$
Thats close!
A different text has
$(r_a, r_b, r_c, r_d, r_e) = (0.21, 0.12, 0.09, 0.13, 0.45)$
Thats NOT close.
Look at

$$p_a q_a + p_b q_b + p_c q_c + p_d q_d + p_e q_e = 0.307$$

Note that in this calculation the large number are multiplied by each other so they result in a large number. Contrast this to:

$$p_a r_a + p_b r_b + p_c r_c + p_d r_d + p_e r_e = 0.196$$

This is much smaller! Consider the following measure:

Definition 3.1 If $\vec{p}$ and $\vec{q}$ are two vectors of probabilities of length n then $d(\vec{p}, \vec{q}) = \sum_{i=1}^{n} p_i q_i$. (For those who have had linear algebra this is a dot product.)

Let $\vec{p}$ be the vector of length 26 for prob of letters in English. The following is known (empirically)

- If T is a long document in English with probability vector $\vec{q}$ then $d(\vec{p}, \vec{q}) \sim 0.68$.
- If T is a shifted document with probability vector $\vec{q}$ then $d(\vec{p}, \vec{q}) \sim 0.32$.
- For other languages there are similar numbers where there is a wide difference between (say) Spanish and Shifted-Spanish.
- Similar numbers probably hold for AFFINE cipher and other ciphers but I have not found anything about that on the Web (possibly ugrad research project!).

With this in mind we now write a program that will, given a text that you know was shifted, decode it. Eve can use it to determine what Alice told Bob! The programs are called IS-ENGLISH? Let $\vec{p}$ be the vector of probs in English.

We will assume that the text we are given is already translated to numbers with $a = 0$, $b = 1$, etc.

IS-ENGLISH?:

1. Input (long) text T (that we assume is a shifted text).
2. Find $\vec{q}$, the vector of probabilities of letters in T (how to do this with just one pass through T is a Homework Assignment).
3. Compute $DOT = d(\vec{p}, \vec{q})$. (how to do this easily is a very easy Homework Assignment).
4. If $DOT \geq 0.66$ then output YES THIS IS ENGLISH. If $DOT \leq 0.34$ then output NO THIS IS NOT ENGLISH. If $0.34 < DOT < 0.66$ then output THIS WAS NOT CODED USING SHIFT! (The parameters 0.66 and 0.34 are used since we only know that English is approx 0.68 and shifts are approx 0.32. If we actually wrote these programs and tested them and used them we might adjust our parameters.)

Note 3.2 There are measures for how close a text is to English. For example, one can use variance. However, using dot product seems to be effective and computationally efficient. We leave it to the reader to work this out rigorously.

How would we change this for affine ciphers? All that changes is the parameters 0.66 and 0.34. So how did we get those?

0.66 is easy: we just did the DOT of real english freq with themselves.

What about 0.34? We did the following:

For $s \in 1, \ldots, 25$ let $F(s)$ be the frequencies in english shifted by s. Then compute $DOT(F(0), F(s))$. The max of these was 0.32.

For Affine we would need to do this for all (a, b) and $ax + b$. More generally, if we have a set of possible codes we can find the correct parameters.

SHIFT-TEXT:

```
1. Input (long) text $T$ (that we assume is a shifted text) and a number $s$. Assume its $t_1 t_2 \cdots t_N$.
2. For $i = 1$ to $N$ $t'_1 = t_1 + s \pmod{26}$.
3. Let $T' = t'_1 t'_2 \cdots t'_N$.
4. Output $T'$
```

DECODE-SHIFT:

```
1. Input (long) text $T$ (that we assume is a shifted text).
2. For $s = 0$ to 25
    (a) $T' =$ SHIFT-TEXT$(T, s)$.
    (b) Compute $b =$IS-ENGLISH$(T')$.
    (c) If $b = YES$ then Output$(T')$ and halt.
    (d) If $b =$THIS IS NOT A SHIFTED TEXT then output THIS IS NOT A SHIFTED TEXT and halt.
3. (If you got to this step then none of the texts were thought to be correct.) Output THIS IS NOT A SHIFTED TEXT and halt.
```