Jan 25, 2016 at 9:27pm

[Quote](https://cicada3301.boards.net/post/16/quote/12)[like](https://cicada3301.boards.net/thread/12/case-shift-independent-text-analysis)

Post Options

**Post by mortlach on Jan 25, 2016 at 9:27pm**

A Case For "Shift Independent" Text Analysis  
  
  
  
When "brute forcing" or running lots of keys it is useful to have a "merit" function for a piece of decrypted text.  
  
The merit function gives a score, as a number, for any piece of text. Better scores mean the text is closer to runeglish,  
  
bad scores indicate nonsense text. Merit functions are of critical importance when using optimization algorithms like  
  
simulated annealing and genetic algorithms.  
  
  
  
A highly effective way to score text is to use n-gram probabilities, I tend to use 4-grams, i.e. groups of 4 letters.  
  
The idea is simple: some combinations of 4 letters are much more common in runeglish than others. For example, "ZZZZ"  
  
will hardly ever occur compared to "BRINGS". If we know the probabilities of these 4-rune combos we can use them in  
  
our merit function.  
  
  
  
One issue (amongst many) when scoring text is that we may only be a simple shift from the correct answer.  
  
Due to the number of runes, 29, and the forward or reverse ordering that gives us 58 'simple shifts' to consider. Naively,  
  
one might apply each of the 58 possible shifts to a piece of text to see if it is a simple shift away from runeglish.  
  
However, I think this is not necessary, and is best illustrated with an example. Take the "A Warning..." page from the  
  
Liber Primus, the first few cipher-text runes are:  
  
  
  
ᚱ ᛝ ᚱ ᚪ ᛗ ᚹ ᛂ ᛁ ᚻ ᛖ ᛁ ᛡ ᛁ ᛗ ᚫ ᚣ ᚹ ᛠ ᚪ ᚫ ᚾ ᚣ ᛖ ᛈ ᛂ ᚫ ᚫ ᛞ  
  
  
  
One way to convert the runes to numbers is to use their relative position in the Gematria Primus. These are the positions  
using the "Forward Gematria" starting with F = 0,  
  
  
  
4, 21, 4, 24, 19, 7, 11, 10, 8, 18, 10, 27, 10, 19, 25, 26, 7, 28, 24, 25, 9, 26, 18, 13, 11, 25, 25, 23  
  
  
  
These are the positions using the Reverse Gematria starting EA = 0,  
  
  
  
24, 7, 24, 4, 9, 21, 17, 18, 20, 10, 18, 1, 18, 9, 3, 2, 21, 0, 4, 3, 19, 2, 10, 15, 17, 3, 3, 5  
  
  
  
When used in our merit function of 4-gram probabilities the above two texts will give different scores, but, we know they  
  
are only a simple shift away from each other. The symmetry of the "shifts, forward and reverse Gematria" necessitates an  
  
"invariant" property of the shifts. By invariant I mean a quantity that does not change, no matter what shift we consider.  
  
One such invariant is the absolute value of the differences between runes position numbers. When Mod 29'd the absolute  
  
difference between successive positions is:  
  
  
  
17, 17, 20, 5, 12, 4, 1, 2, 10, 8, 17, 17, 9, 6, 1, 19, 21, 4, 1, 16, 17, 8, 5, 2, 14, 0, 2  
  
  
  
for BOTH the above shifts, (and all the other 56 possible shifts as well).  
  
  
  
This fact is extremely useful. (I cannot emphasize enough how useful invariants are for analysis purposes).  
  
  
  
If instead of using 4-gram probabilities we use the "absolute difference 4-grams" then our merit function will 'always' take  
  
into account any simple shift of a given a text. Calculating these probabilities is trivial and there are many guides, e.g.  
  
  
  
[practicalcryptography.com/cryptanalysis/text-characterisation/quadgrams/](http://practicalcryptography.com/cryptanalysis/text-characterisation/quadgrams/)  
  
  
  
This technique has 2 practical purposes I can think of immediately:  
  
  
  
1. No need to brute force all 58 shifts,  
  
2. When considering sequences as streams you don't have to consider the sequence -1, or -2, or +1,+2,+3 etc... making  
  
the (in my opinion) messy primes-1 sequence for page 56 not an issue.  
  
  
  
Obviously, when a good score is found in a optimization procedure, you will still have to go back and find the optimum shift  
  
to recover the runeglish, but the hard work has already been done ;-)  
  
(Further work in this area will show that keys can also be declared in a shift independent fashion. Coming soon ?)