CRYPTOLOGIA July 1991 Volume XV Number 3

VOWEL IDENTIFICATION: AN OLD (BUT GOOD) ALGORITHM

Jacques B. M. Guy

ADDRESS: Telecom Australia Research Laboratories, P O Box 249, Clayton 3168 AUSTRALIA.

ABSTRACT: A very fast and accurate algorithm for identifying vowels and consonants in plaintext or in a simple substitution cipher.

KEYWORDS: Vowel identification, simple substitution cipher.

In the October 1990 issue of *Cryptologia* Caxton Foster [1] proposes an algorithm to identify which letters of a simple substitution cipher are vowels. A very fast algorithm which does precisely that was discovered almost 30 years ago by a Soviet researcher by the name of B. V. Sukhotin. Unfortunately, Sukhotin published only in Russian and was ever translated only into French. So here is a world's first: Sukhotin's algorithm in English[1]

Sukhotin had observed that vowels tend to occur next to consonants rather than next to vowels. And indeed if they did not this sentence would look more or less like: and nddiee if they idd nto iths sntncee ouwld lkoo oemr ro elss lkie. Skhtnuoi's lgrthmaoi...er...Sukhotin's algorithm is wholly based on that observation.

Say we have this text: "SAGITTA" and we want to know which letters are vowels and which consonants. To make things just a little bit more interesting (and this ought to appeal to decipherers of the Phaistos disk), let us also suppose that it was written around a plate, so that there is no way of being sure

[1] Well, not Sukhotin's algorithm word for word, but it boils down to exactly the same; it only gets there faster, with less number shuffling. Greg Mellen points out to me that Helen Fouche Gaines tackled the same problem by means of a very similar-looking matrix in her *Elementary Cryptanalysis* (pp. 78-85 of the 1956 Dover edition), and credited the method to M. E. Ohaver (1933), so that Sukhotin did not "discover" the algorithm described here but simplified the method and formalized it. I have had Gaines's book for many years and have read it through. Yes, the construction of the matrix is similar (save that it is not symmetrical), but the method followed is very, very different indeed... well, at least it seems to me. Thus for instance the first task tackled is the tentative identification of the digram "th", followed next by that of "he" and "ha". Ohaver's method also relies on a previous knowledge of the digram frequencies of the language in which the message is written. Sukhotin, on the other hand, assume a state of complete ignorance about the language, except that the writing system is alphabetical. For those reasons, I would still hold that Sukhotin's algorithm is wholly original.

where it starts (and ends), even in which direction it ought to be read. It could be "SAGITTA" or "ATTIGAS", "AGITTAS" or "SATTIGA", "GITTASA" or "ASATTIG" etc. for all we know.

Step 1. Count the number of times each letter of the text is in contact with another. Thus "G" is in contact with "I" once (and "I" with "G" once), "A" with "S" twice (and "S" with "A" twice; remember: the text is written in a circle[2], and so on. You end up with a symmetrical frequency matrix:

	S	A	G	I	T
S	0	2	0	0	0
A	2	0	1	0	1
G	0	1	0	1	0
I	0	0	1	0	1
T	0	1	0	1	2

Step 2. Fill the main diagonal with zeroes:

	S	A	G	I	T
S	0	2	0	0	0
A	2	0	1	0	1
G	0	1	0	1	0
I	0	0	1	0	1
T	0	1	0	1	0

Step 3. Sum the rows, and assume all letters are consonants:

	S	A	G	I	T	Sum	
S	0	2	0	0	0	2	consonant
A	2	0	1	0	1	4	consonant
G	0	1	0	1	0	2	consonant
I	0	0	1	0	1	2	consonant
T	0	1	0	1	0	2	consonant

Step 4. Find the consonant with the highest sum greater than zero. That consonant was really a vowel all along:

	S	A	G	I	T	Sum	
S	0	2	0	0	0	2	consonant
A	2	0	1	0	1	4	it's a vowel!!!
G	0	1	0	1	0	2	consonant
I	0	0	1	0	1	2	consonant
T	0	1	0	1	0	2	consonant

[2]There is no harm done if we know where the text starts and ends, in which direction it reads, and where word separations are: just adjust the count of the number of times a letter occurs next to another accordingly.

Step 5. Now subtract from the sum of the row of each consonant twice the number of times it occurs next to the new-found vowel. For instance, "S" occurs twice next to "A", so the sum of its row becomes 2 - 4 = -2. Do the same for the other "consonants" ("G", "I", and "T"):

	S	A	G	I	T	Sum	
S	0	2	0	0	0	-2	consonant
A	2	0	1	0	1	4	vowel
G	0	1	0	1	0	0	consonant
I	0	0	1	0	1	2	consonant
T	0	1	0	1	0	0	consonant

Now continue from Step 4 until you find no more new vowels. Thus for instance here, "I" being the consonant with the highest sum greater than zero, it becomes a vowel at Step 4:

	S	A	G	I	T	Sum	
S	0	2	0	0	0	-2	consonant
A	2	0	1	0	1	4	vowel
G	0	1	0	1	0	0	consonant
I	0	0	1	0	1	2	it's another vowel!!!
T	0	1	0	1	0	0	consonant

Then Step 5 leaves us with:

	S	A	G	I	T	Sum	
S	0	2	0	0	0	-2	consonant
A	2	0	1	0	1	4	vowel
G	0	1	0	1	0	-2	consonant
I	0	0	1	0	1	2	vowel
T	0	1	0	1	0	-2	consonant

Since there are no consonants left with a sum greater than zero, the process stops there.

Simple, wasn't it? But was there any logic in all that number-shuffling? Yes, there was. As Sukhotin did not explain the rationale behind his algorithm you have to piece it together yourselves, though.

What the column headed "Sum" contains at every step is in fact the difference between the number of times a letter is found next to a consonant and the number of times it is found next to a vowel. At Step 4 we pick the letter with the highest difference and call it a vowel (logical, since the fundamental assumption is that vowels occur next to consonants rather than to vowels). Then, at Step 5, we

recompute those differences again (only taking quite a tremendous computational shortcut). Eventually no letter is left which occurs more often next to consonants than next to vowels (i.e. such that Sum> 0) and the process stops there[3]

REFERENCES

1. Foster, Caxton. 1990. Vowel Distribution as a Clue to Vowel Identification. *Cryptologia*. 14(4): 355-362.

2. Gaines, Helen F. 1956. *Cryptanalysis*. New York: Dover. (First edition 1939).

3. Sukhotin, B. V. 1962. Eksperimental'noe vydelenie klassov bukv s pomoshch'ju elektronnoj vychislitel'noj mashiny. *Problemy strukturnoj lingvistiki*. 234: 198-206.

4. Sukhotin, B. V. 1973. Methode de dechiffrage, outil de recherche en linguistique. *T. A. Informations*. 2: 3-43

5. Tretiakoff, A. 1976. Identification des phonemes dans la langue ecrite par un critere d'information maximum. *Papers in Computational Linguistics*. (Ferenc Papp & Gyorgy Szepe eds.) 403-408. Paris: Mouton.

ACKNOWLEGEMENTS

The permission of the Executive General Manager, Telecom (Australia) Research Laboratories, to publish this material is gratefully acknowledged.

BIOGRAPHICAL SKETCH

After doing a PhD in linguistics on an obscure dialect of Espiritu Santo (Vanuatu, then called New Hebrides), Jacques Guy learned computer programming, thinking it would provide him with a better meal ticket. As it turned out, he stayed on for 12 years in the Department of Linguistics of the Australian National University in Canberra, teaching himself in the process (and in this order)

[3]In some cases (very rare), letters which are properly consonants get misidentified as vowels because the algorithm stops just a little bit too late. In other cases a consonant or two can get misclassified as vowels because there is precious little correlation between pronunciation and spelling, so that the "consonants" and "vowels" of the alphabet correspond only loosely to the "consonants" and "vowels" of pronunciation. English is a particularly nasty instance of such a case: "Y" is a vowel in "JELLY", but is it in "YELLOW"? And what about in "MY"? Isn't it more like two vowels? How about "K", "G", and "H" which are all silent in "KNIGHT"? And "U" which is just as silent in "LAUGH" as "K", "G" and "H" in "KNIGHT"? In spite of those anomalies Sukhotin's algorithm still works fairly well on English: it misidentifies only "T" as a vowel (and then not always). The reason is the very high frequency of the digram "TH" which is really one single consonant, not two.

ALGOL, Basic, Simula 67, statistical methods, PL/I, C, and Pascal. He now works in the Department of Artificial Intelligence of the research laboratories of Telecom Australia. The author is also interested in the Voynich manuscript and in the decipherment of the tablets of Easter Island.

"I think you could safely say that the artisan hit his thumb at that point."